

**CARACTERIZACIÓN DE FUENTES Y SUMIDEROS
DE LA INTRANET DE LA UIS**

JORGE ALBERTO GÓMEZ MARTÍNEZ

**ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
UNIVERSIDAD INDUSTRIAL DE SANTANDER
BUCARAMANGA
2009**

**CARACTERIZACIÓN DE FUENTES Y SUMIDEROS
DE LA INTRANET DE LA UIS**

JORGE ALBERTO GÓMEZ MARTÍNEZ

**Investigación para optar al título de
Magíster en Ingeniería Electrónica**

**Director
PhD. Oscar Gualdrón González
Ingeniero de Sistemas**

**ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
UNIVERSIDAD INDUSTRIAL DE SANTANDER
BUCARAMANGA
2009**

Dedico este trabajo a mis Padres Amparo Martínez Peña y Antonio Gómez Serrano, por su apoyo incondicional.

Jorge Alberto Gómez Martínez

AGRADECIMIENTOS

El autor expresa sus agradecimientos:

Al doctor Oscar Gualdrón por transmitir sus conocimientos, experiencias, consejos y por guiarme en esta tarea.

TABLA DE CONTENIDO

PLANTEAMIENTO DEL PROBLEMA Y ALCANCE.....	14
OBJETIVOS.....	15
1. INTRODUCCIÓN.....	16
1.1 Estructura del documento.....	17
2. HERRAMIENTAS DE SIMULACIÓN DE TRÁFICO.....	18
2.1 Importancia de la simulación de la Intranet UIS.....	18
2.2 Herramientas de simulación.....	19
2.3 Network Simulator.....	27
2.3.1 Requerimientos del Sistema.....	27
2.3.2 ¿Qué se puede hacer con la herramienta?.....	28
2.3.3 Interfaz de usuario.....	29
2.3.4 Resultados con ns2.....	29
2.3.5 ¿Por qué ns2?.....	30
2.4 Modelo de pruebas.....	31
3. AUTOSIMILITUD.....	32
3.1 Concepto de Autosimilaridad [1, 2, 3, 4, 6, 22, 23].....	32
3.2 El parámetro de Hurst.....	35
3.3 Formas de detectar la Autosimilaridad.....	35
3.3.1 Dependencia a largo plazo [1].....	36
3.3.2 Decaimiento lento de la varianza [1].....	38
3.3.3 Rango reescalado o R/S [7].....	39
3.3.4 Periodograma [1, 2].....	41
3.3.5 Estimador de Whittle [1, 2].....	41
3.3.6 Otros.....	43
4. RESULTADOS EN LA INTRANET UIS.....	44
4.1 Selección de las características del tráfico.....	44
4.2 Pruebas sobre la red.....	46
4.3 Autosimilitud de la red.....	48
4.4 Observaciones de la red.....	60
4.5 Aleatoriedad en la red.....	61

4.5.1 Prueba Kolmogorov-Smirnov (K-S) [18].....	63
4.5.2 Prueba Chi-cuadrado (Chi 2) [18]	64
5. SIMULACIÓN DE LA RED	65
5.1 Escenario de simulación y ajustes	65
5.2 Autosimilitud de la simulación	69
5.3 Aleatoriedad de la simulación.....	73
5.4 Observaciones de la simulación.....	74
6. CONCLUSIONES	76
BIBLIOGRAFÍA.....	78

LISTA DE FIGURAS

Figura 1 Tráfico red UIS contra tráfico generado con un simulador de tráfico.....	20
Figura 2 Interfaz gráfica de AdventNet 6 [12]	21
Figura 3 Interfaz gráfica de la herramienta FLAN. [8].....	22
Figura 4 Interfaz gráfica de Kiva. [8].....	23
Figura 5 Interfaz gráfica del Network Simulator. [13].....	24
Figura 6 Interfaz gráfica del OMNeT++. [15]	25
Figura 7 Interfaz Gráfica del OpNet IT Guru. [14].....	25
Figura 8 Interfaz gráfica del Packet Tracer [16]	26
Figura 9 Interfaz gráfica de Shunra VE Desktop. [17]	27
Figura 10 Modelo de pruebas.	31
Figura 11 Trazas de tráfico tomadas del core switch de la UIS en tres escalas de tiempo....	33
Figura 12 Poisson contra auto-similitud. [4]	34
Figura 13 Triángulo de Sierpinski, utilizado en telecomunicaciones. [10].....	35
Figura 14 Función de autocorrelación.	37
Figura 15 Decaimiento lento de la varianza.....	38
Figura 16 Rango Reescalado.....	40
Figura 17 Periodograma.	42
Figura 18 Esquema de evaluación.....	45
Figura 19 Topología de la Intranet de la Universidad Industrial de Santander.	46
Figura 20 Resultados prueba 1, para el core switch día 1.....	49
Figura 21 Resultados prueba 1, para el core switch día 2.....	50
Figura 22 Resultados prueba 1, para el core switch día 3.....	51
Figura 23 Resultados prueba 1, para el core switch día 4.....	52
Figura 24 Resultados prueba 1, para el core switch día 5.....	53
Figura 25 Resultados prueba 2, para el core switch día 1.....	54
Figura 26 Resultados prueba 2, para el core switch día 2.....	55
Figura 27 Resultados prueba 2, para el core switch día 3.....	56
Figura 28 Resultados prueba 2, para el core switch día 4.....	57
Figura 29 Resultados prueba 2, para el core switch día 5.....	58
Figura 30 Resultados prueba 2, para el core switch promediado.....	59
Figura 31 Correlación de los servidores contra el core switch.	60
Figura 32 Función de densidad de probabilidad de los servidores Cóndor y Dodo.	63
Figura 33 Escenario de simulación en ns2.....	65
Figura 34 Resultados prueba 1, para el tráfico agregado del nodo 0.....	70
Figura 35 Resultados prueba 2, para el tráfico agregado del nodo 0.....	72
Figura 36 Histogramas de los nodos 3 y 4.	75
Figura 37 Correlación de los nodos servidor contra el nodo de tráfico agregado.....	75
Figura 38 Edición de las preferencias de Wireshark.....	81
Figura 39 Opciones de captura en Wireshark.	82
Figura 40 Pantalla para exportar a *.txt.....	82
Figura 41 Pantalla principal de Selfis.....	83

LISTA DE TABLAS

Tabla 1 Ventajas y desventajas de las herramientas de simulación de redes. [8].....	19
Tabla 2 Resumen de pruebas realizadas al Core Switch de la UIS.....	47
Tabla 3 Resultados prueba 1, para el core switch día 1.	49
Tabla 4 Resultados prueba 1, para el core switch día 2.	50
Tabla 5 Resultados prueba 1, para el core switch día 3.	51
Tabla 6 Resultados prueba 1, para el core switch día 4.	52
Tabla 7 Resultados prueba 1, para el core switch día 5.	53
Tabla 8 Resultados prueba 2, para el core switch día 1.	54
Tabla 9 Resultados prueba 2, para el core switch día 2.	55
Tabla 10 Resultados prueba 2, para el core switch día 3.	56
Tabla 11 Resultados prueba 2, para el core switch día 4.	57
Tabla 12 Resultados prueba 2, para el core switch día 5.	58
Tabla 13 Resultados prueba 2, para el core switch promediado.....	59
Tabla 14 Resultados probabilísticos: Condor, Dodo, Core switch.	62
Tabla 15 Descripción de los nodos del escenario de simulación.	66
Tabla 16 Resultados prueba 1, para el tráfico agregado del nodo 0.....	70
Tabla 17 Resultados prueba 2, para el tráfico agregado del nodo 0.....	71
Tabla 18 Promedios autosimilitud simulados contra empíricos prueba 2.....	72
Tabla 19 Resultados probabilísticos nodos: 0, 3 y 4.....	74

LISTA DE ANEXOS

ANEXO A: CONFIGURACIÓN DEL WIRESHARK.....	80
A.1. Adquisición de datos con Wireshark	80
A.2 Análisis con la herramienta Selfis	81
ANEXO B: CÓDIGOS PARA LAS PRUEBAS.....	84
B.1 Código para la prueba 1 (MATLAB).....	84
B.2 Código para la prueba 2 (MATLAB).....	87
ANEXO C: CÓDIGOS DEL SIMULADOR.....	91
C.1 Simulador Intranet UIS.....	91
C.2 Código para obtener el grano deseado del simulador	94

RESUMEN

TÍTULO

CARACTERIZACIÓN DE FUENTES Y SUMIDEROS DE LA INTRANET UIS¹

AUTOR

Jorge Alberto Gómez Martínez²

PALABRAS CLAVE

Aleatoriedad, auto similaridad, dependencia a largo plazo, estimador de Whittle, parámetro de Hurst, periodograma, prueba de bondad de ajuste.

DESCRIPCIÓN

Este trabajo de maestría propone el análisis de un segmento de la Intranet UIS, con el fin de modelar su comportamiento y posteriormente obtener resultados cercanos mediante el uso de simuladores de tráfico.

Para lograr esto, se recogen datagramas de servidores con protocolos comúnmente utilizados como el SMTP (*Simple Mail Transfer Protocol*), HTTP (*Hypertext Transfer Protocol*), entre otros. Con dichos datagramas se determina el grado de auto-similitud y las características probabilísticas que presenta el segmento de red, utilizando diferentes pruebas para cada caso.

Las pruebas se realizan a cinco días de tráfico, que incluye: cálculo del parámetro de Hurst mediante el estimador de Whittle y periodograma para diferentes escalas de tiempo, para cada servidor y para el tráfico agregado de todos los servidores; una comparación de correlación de cada servidor contra el tráfico agregado; pruebas de bondad de ajuste para obtener la función de densidad de probabilidad del tráfico de algunos servidores y del tráfico agregado.

Una vez obtenido el modelo del segmento de red, se utiliza una herramienta de simulación para conseguir un ambiente de tráfico similar al real.

Se presenta además en este documento: algunos simuladores de tráfico, se hace especial énfasis en el Network Simulator 2 con el cual se desarrolló la simulación de esta investigación; distintas técnicas para calcular el parámetro de Hurst; descripción para realizar pruebas de bondad de ajuste.

¹ Proyecto de Maestría.

² Facultad de Físico-Mecánicas. Escuela de ingeniería eléctrica, electrónica y telecomunicaciones.
Director Oscar Gualdrón Gonzalez

ABSTRACT

TITLE

SOURCE AND SINK CHARACTERIZATION IN UIS INTRANET³

AUTHOR

Jorge Alberto Gómez Martínez⁴

KEYWORDS

Randomness, self-similarity, long range dependence, Whittle estimator, Hurst parameter, periodogram, goodness of fit test.

DESCRIPTION

This work consists of an analysis of a traffic segment from UIS Intranet, heading to model its behavior and later get similar results by using a traffic simulation tool.

To achieve this, datagrams are collected from servers with commonly used protocols such as SMTP (Simple Mail Transfer Protocol) and HTTP (Hypertext Transfer Protocol). The datagrams are then used to determine the degree of self-similarity of the net segment and its probabilistic characteristics, different tests are used for each case.

The tests are done over five traffic days, which include: Hurst parameter calculation using Whittle estimator and periodogram for different time scales, at each server and the aggregated traffic of all servers; a correlation comparison of each server versus the aggregated traffic; goodness of fit test to get the traffic probability density function of some servers and the aggregated traffic.

Once the net segment model is obtained, a simulation tool is used to get a similar traffic environment to the real net traffic.

In this document there are also introduced: a few traffic simulation tools, special emphasis is made on the Network Simulator 2 which is used to develop the simulation for this investigation; different Hurst parameter calculation methods; description about goodness of fit tests.

³ Master of science proyect.

⁴ Faculty of Physics and Mechanics. School of electrical, electronic and telecommunication engineering. Director Oscar Gualdrón Gonzalez.

PLANTEAMIENTO DEL PROBLEMA Y ALCANCE

La comprensión analítica del comportamiento de las redes ha avanzado más lentamente que la misma tecnología que la conforma. Es necesario empezar a comprender cómo se logra llegar a un modelo matemático que permita caracterizar el tráfico de la intranet universitaria y que facilite la evolución de la misma.

La Intranet actual presenta algunos problemas de latencia y pérdida de datos, por lo cual su desempeño no es adecuado bajo circunstancias críticas. Es posible solucionar parte de estos problemas utilizando el modelo de la red en una herramienta de simulación, que permita experimentar con la misma de forma rápida y de bajo presupuesto.

Muchas investigaciones de este tipo se han realizado en otras redes y llegan a conclusiones importantes en cuanto a su comportamiento. De acuerdo con dichas investigaciones, las redes pueden presentar fenómenos de auto-similaridad, propiedad característica de un modelo fractal, cuyo conocimiento permite evaluar, controlar y predecir con resultados razonablemente aproximados el rendimiento de la red. Sin embargo, encontrar un modelo de este tipo requiere de un largo estudio del tráfico de la red, en éste caso particular la Intranet UIS.

OBJETIVOS

OBJETIVO GENERAL

Caracterizar fuentes y sumideros de sectores específicos de la UIS como insumo base para la construcción de un modelo del comportamiento del tráfico de la red.

OBJETIVOS ESPECÍFICOS

- Seleccionar características relevantes de tráfico en fuentes y sumideros que serán objeto de observación en el trabajo.
- Proponer un modelo aproximado del comportamiento de los sectores estudiados en la red.
- Implementar el modelo propuesto en una herramienta de simulación de tráfico.
- Contrastar los resultados obtenidos en el simulador con el comportamiento real de la red.

1. INTRODUCCIÓN

Con el avance de la computación y de las telecomunicaciones, evolucionó a su vez una tecnología para intercambiar información conocida actualmente como Internet, que tiene sus orígenes a principios de los 1960s en Estados Unidos, debido a la necesidad de transmitir información militar en forma segura en caso de un ataque nuclear. La solución era segmentar un mensaje largo en muchos pedazos más pequeños, también conocidos como paquetes. Gracias a esta grandiosa idea de la corporación RAND, el Dr. Metcalfe invento la Ethernet a finales de los 1970s. [4]

Internet evolucionó rápidamente en los últimos 30 años, y es ahora una de las fuentes de información más consultada con un impacto de nivel mundial. Sin embargo, su entendimiento no ha evolucionado a la misma velocidad. El diseño de redes robustas y servicios a través de las mismas se ha convertido en la regla de Internet. Lograr este objetivo requiere el entendimiento de las características del tráfico de datos que circula por las mismas, dicho entendimiento es crucial para el mejoramiento del servicio.

En 1990 Will Leland y Daniel Wilson [3] comenzaron a proponer modelos estadísticos sobre el comportamiento de las redes LAN, concluyendo que los modelos tradicionales aplicados en comunicaciones no se adaptaban muy bien a las redes LAN. Investigaciones posteriores encontraron que los modelos fractales aplicados no en espacio si no en el tiempo, permitían modelar aproximadamente el comportamiento de dichas redes, a éste comportamiento en la dimensión del tiempo se le conoce comúnmente como autosimilaridad (self similarity) [6].

El comportamiento autosimilar se debe a la distribución del tamaño de los paquetes, la interacción humana con el medio, la dinámica de la red y la dependencia a largo plazo (LRD). Los problemas que esto ocasiona se reflejan directamente en el análisis y el diseño de las redes, y como consecuencia final los conocimientos previos sobre los cuales se desarrollaron estos sistemas ya no son válidos.

Desde aquí nace el interés por comenzar a entender matemáticamente el comportamiento de un recurso tan explotado que es la Intranet de la UIS, y de acuerdo a los resultados encontrados, proponer nuevas investigaciones para el mejoramiento de la misma.

Este trabajo de post grado propone mediante la recolección de datagramas en algunos de los servidores de la UIS, encontrar el grado de auto similaridad de la

red, obtener su modelo probabilístico y finalmente llegar a resultados similares en una herramienta de simulación de tráfico.

1.1 Estructura del documento

El documento está dividido en cuatro capítulos, cuyo contenido se resume a continuación:

El capítulo 2, explica la importancia de utilizar herramientas de simulación de tráfico, expone un compendio de diferentes herramientas con sus características y muestra el modelo de pruebas utilizado con la herramienta seleccionada para hacer la simulación.

El capítulo 3, presenta el concepto de auto-similitud, como se detecta mediante los diferentes estimadores y se explica el parámetro de Hurst, el cual se utiliza para indicar el grado de auto-similitud.

El capítulo 4, muestra los resultados del parámetro de Hurst en el segmento de red estudiado. Además, se encuentran los resultados probabilísticos y la función de densidad de probabilidad calculada mediante una prueba de ajuste de bondad para dicho segmento de red.

El capítulo 5, contiene la simulación del tráfico con la herramienta software seleccionada, cuyos resultados son comparados con los encontrados en el capítulo 4.

El anexo A, da una guía de la metodología a seguir con el software Wireshark [20] y Selfis [19] para encontrar el parámetro de Hurst.

El anexo B, contiene los códigos en Matlab utilizados en la segmentación de los datagramas obtenidos para realizar las pruebas de análisis.

El anexo C, presenta el código en lenguaje tcl para simular la red con la herramienta Network Simulator 2.

2. HERRAMIENTAS DE SIMULACIÓN DE TRÁFICO

Las herramientas de simulación en general han sido de gran valor para el análisis y comprensión de sistemas complejos. Los simuladores van desde lo físico, como lo hacen los astronautas en el agua para tener un ambiente similar al que experimentarían cuando se encuentran en el espacio; hasta simuladores virtuales, que a diferencia de los anteriores no hay interacción humana, y se sustentan en modelos matemáticos detallados, sin embargo, en lo que sí coinciden es que generan un ambiente similar al que ocurriría en una situación real.

Para los simuladores virtuales es fundamental el diseño del modelo y la herramienta software. El modelo sencillamente debe ser tan similar al original como sea posible, teniendo en cuenta los factores principales a los cuales se somete el sistema real; la herramienta debe ser lo suficientemente flexible para poder manipular dichos factores.

Al final el objetivo fundamental de los simuladores es comprender y/o evaluar el comportamiento del objeto real simulado, cuyos resultados deben ser lo suficientemente aproximados a los reales para extrapolarlos, y como consecuencia ahorrar tiempo y dinero al poder solucionar problemas en forma virtual.

2.1 Importancia de la simulación de la Intranet UIS

La Intranet UIS es una herramienta de gran valor en la comunidad, y su alta demanda obliga a aumentar los recursos de la misma, por ejemplo el ancho de banda es un recurso escaso y costoso, sobre todo cuando se trata de redes públicas en las cuales el servicio es pago. El análisis del tráfico permite administrar adecuadamente este recurso, y además, proponer cambios de topología o esquemas de control para ello. Estos problemas no son un fenómeno aislado, y tanto para la Intranet UIS como para el resto de Internet las causas y sus consecuencias son similares.

El primer factor que ocasiona el incremento de dichos recursos, es el número de nodos de la red que aumenta en forma permanente. Otra causa es la evolución de los servicios ofrecidos por las aplicaciones, cada nueva aplicación tiene características diferentes que contribuyen a aumentar el tráfico. Adicionalmente a esto, se suma la tecnología inalámbrica que facilita a usuarios ocasionales el acceso a la red. Y por supuesto el aumento de la demanda informática.

Conocidas las causas de la congestión del tráfico, es vital señalar que la solución no siempre es incrementar los recursos de la red, pues no importa cuantos recursos posea el usuario, nunca serán suficientes y siempre habrá usuarios que saturan la red con aplicaciones nuevas o descargas de archivos.

Es aquí donde la simulación del tráfico juega un papel fundamental. Primero, permite el entendimiento detallado de cómo se comporta el medio, ofreciendo al investigador la posibilidad de manipular y experimentar en forma virtual sobre la red de datos, y segundo, permite la experimentación con la misma sin tener que medir el comportamiento en campo, sino en el esquema de simulación. Este proceso conduce a un ahorro monetario al poder reconocer las fallas del sistema más rápidamente, y a futuro podría conducir a limitar el uso de aplicaciones que no solo congestionan la red, sino que a su vez no están contemplados dentro de los servicios que debe ofrecer.

En la tabla 1 se listan algunas ventajas y desventajas de la utilización de herramientas de simulación de redes.

Tabla 1 Ventajas y desventajas de las herramientas de simulación de redes. [8]

Ventajas	Desventajas
<ul style="list-style-type: none"> • Facilita el análisis de tráfico • Permite evaluar el desempeño de las redes frente a la implementación de nuevas tecnologías. • Representa un ahorro sustancial de costos. • Permite la manipulación de la red, efectuar cambios y experimentar virtualmente. • Complementa los conocimientos de análisis de redes. • Mejora la planeación y el diseño de nuevas redes y servicios de telecomunicaciones. 	<ul style="list-style-type: none"> • La veracidad de los resultados de la simulación depende directamente de la herramienta utilizada. • Los resultados de una simulación son únicamente una aproximación del sistema real. • Dificultad para considerar todos los parámetros. • La validación del modelo se realiza con base en fundamentos teóricos.

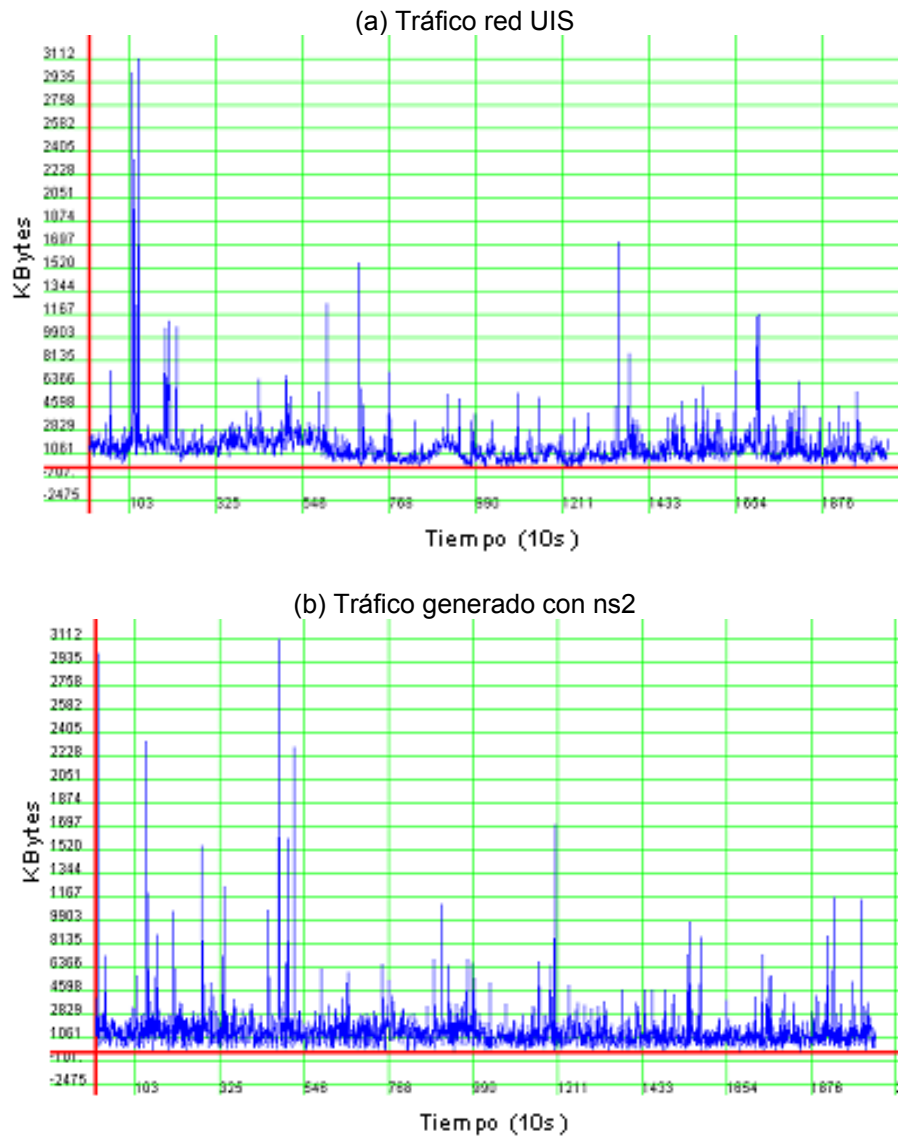
2.2 Herramientas de simulación

Muchas investigaciones se han basado en crear modelos de tráfico más precisos y similares al comportamiento real de una red y estas a su vez evolucionaron en herramientas de evaluación de tráfico.

A continuación en la figura 1 se muestran trazas de tráfico real y trazas de tráfico artificial. En (a), se puede observar el tráfico real medido en la Intranet de la UIS, a la derecha. En (b), se representa el tráfico artificial generado por una herramienta

de simulación de tráfico. Si bien el tráfico en (b) es en teoría aproximado al tráfico en (a), solo un análisis probabilístico nos podrá decir si realmente son similares o no. El éxito de una simulación depende del modelo elegido que representa la situación real. Ver capítulo 4.

Figura 1 Tráfico red UIS contra tráfico generado con un simulador de tráfico.



Existe una gran variedad de aplicaciones para simular redes, algunas de ellas comerciales otras del tipo *Open Source*. Para poder seleccionar una herramienta acorde a la investigación se tomaron algunos de los simuladores más conocidos entre los cuales se encuentran: AdventNet, FLAN, Kiva, Network Simulator, OMNeT++, OpNet IT Guru, Packet Tracer, y Shunra VE Desktop, entre otros; de cada uno se hablará a continuación.

AdventNet [11]

Es una herramienta comercial y específica para redes de datos.

La herramienta de simulación AdventNet fue diseñada para el entrenamiento y demostración de aplicaciones de gestión de redes. Brinda además un editor de topología para establecer interconexiones a través de routers, switches y otros dispositivos de red.

La herramienta permite el agregado masivo de dispositivos con una única dirección IP y puerto, la modificación de las propiedades de los dispositivos como: dirección IP, número de puerto, valores MIB5, modelado avanzado de conducta de agentes y redes, configuración de solicitudes y respuestas. La figura 2 muestra la interfaz gráfica de AdventNet con una red sencilla.

Es un software de propietario.

Figura 2 Interfaz gráfica de AdventNet 6 [12]



FLAN [8]

F- Links and Nodes es un software desarrollado en Java, y es un simulador *open source* de propósito general orientado a objetos.

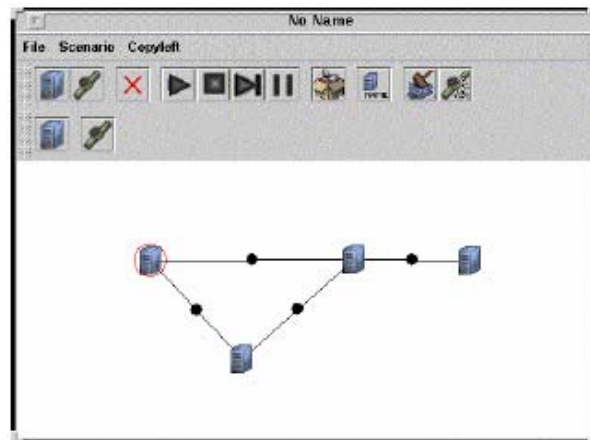
⁵ MIB: Management information base (base de información de administración).

Con Java se pueden crear y configurar nuevos dispositivos, aplicaciones, protocolos, entre otros; al permitir la manipulación de sus librerías.

Este software permite crear y configurar nuevos dispositivos, aplicaciones o protocolos de red, inclusive se pueden realizar modificaciones en su código fuente. FLAN permite el diseño, la construcción, y la prueba de una red de comunicaciones. La figura 3 muestra la interfaz gráfica de FLAN.

FLAN es un software multiplataforma, esta es su principal ventaja, su desventaja es que no simula muy bien redes grandes, y además requiere conocimientos básicos de programación en Java para dar mayor profundidad a las aplicaciones del software.

Figura 3 Interfaz gráfica de la herramienta FLAN. [8]



Kiva [8]

Al igual que FLAN, Kiva está diseñado en Java, y se creó principalmente para simular el comportamiento del protocolo IP, más específicamente para el estudio del tratamiento y enrutamiento de los datagramas en una red. Kiva es un software *open source*. En la figura 4 se puede observar la interfaz gráfica de Kiva.

Su principal desventaja es que necesita la descarga de múltiples archivos para su instalación.

Figura 6 Interfaz gráfica del OMNeT++. [15]

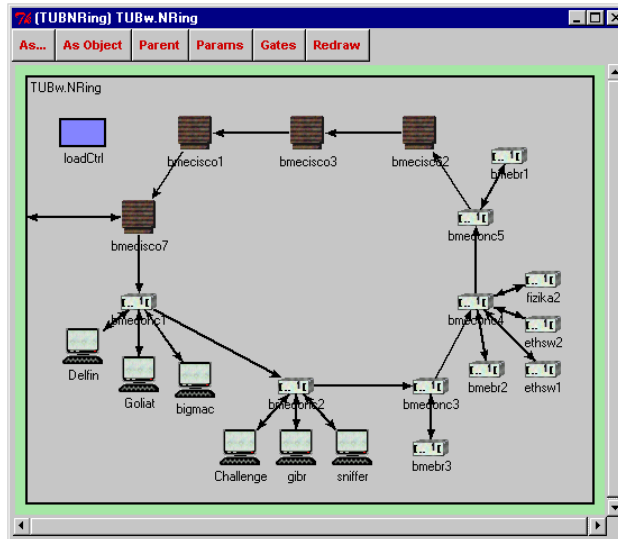
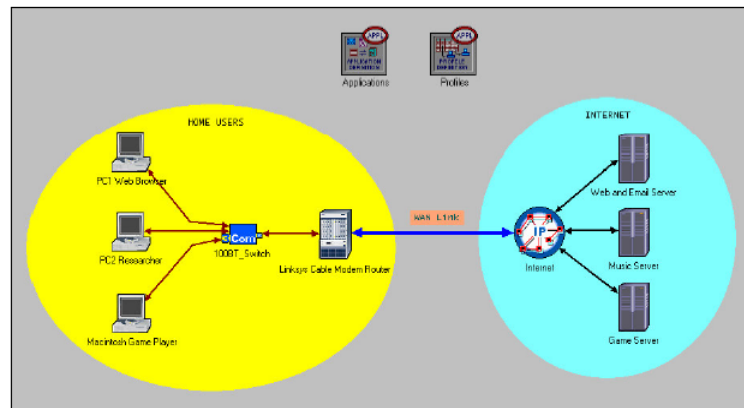


Figura 7 Interfaz Gráfica del OpNet IT Guru. [14]



Packet Tracer [8]

Es un software muy completo específico para simular redes, diseñado por Cisco Systems para uso comercial.

El Packet Tracer permite realizar diagramas de red complejos, configurar routers, switches, protocolos, VLANs⁶, ACLs⁷, crear topologías de red, configurar

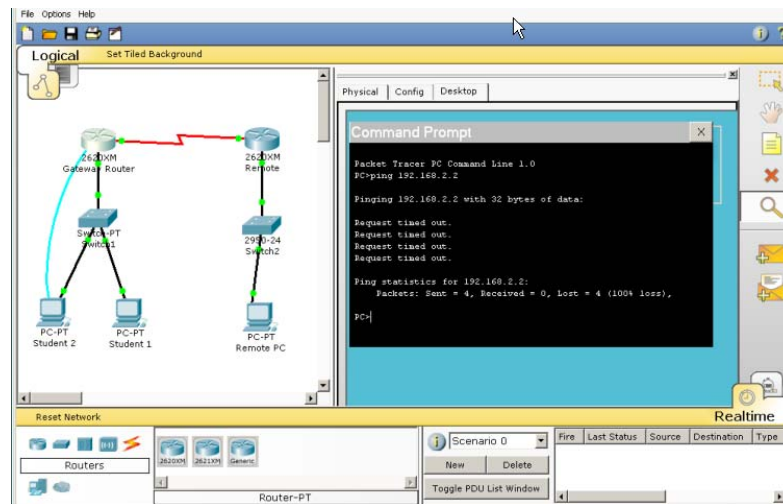
⁶ VLAN: Virtual LAN (LAN virtual)

⁷ ACL: Access control list (lista de control de acceso)

dispositivos, detección y corrección de errores, etc.; todo desde su interfaz gráfica. Ofrece adicionalmente el análisis de acuerdo a la capa de modelo OSI que interviene en la simulación de cada proceso. La figura 8 muestra la interfaz gráfica del Packet Tracer con una red sencilla.

Esta herramienta tiene el propósito de ser utilizada como un producto educativo para conocer la línea de los dispositivos Cisco, para practicar y aprender con ellos.

Figura 8 Interfaz gráfica del Packet Tracer [16]

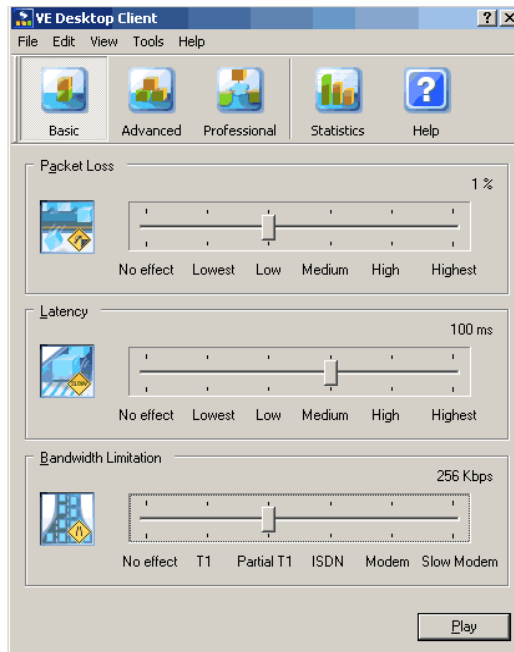


Shunra VE Desktop [17]

Esta herramienta de simulación es específica para la simulación de redes, y es de propietario. Está diseñada para probar el desempeño de aplicaciones de una red. Utiliza parámetros de latencia, fluctuaciones, ancho de banda y pérdidas. También presenta reportes y gráficas en tiempo real de recursos de red utilizados por las aplicaciones activas del PC, incluye medidas de desempeño y tiempos de transferencia. La figura 9 muestra la interfaz gráfica de Shunra VE Desktop.

El software es bastante sencillo y limitado para labores investigativas.

Figura 9 Interfaz gráfica de Shunra VE Desktop. [17]



2.3 Network Simulator

Ns es un simulador de eventos discretos orientado a la investigación de redes. Ns provee soporte sustancial para la simulación TCP, enrutamientos, y protocolos multicast sobre redes cableadas e inalámbricas (locales y satelitales).

Ns comenzó como una variante del REAL Network Simulator en 1989, y ha evolucionado sustancialmente con el paso de los años. En 1995 DARPA ayudó al desarrollo de ns a través del proyecto VINT en LBL, Xerox PARC, UCB y USC/ISI. Actualmente el desarrollo de ns es patrocinado por DARPA con SAMAN y a través de NSF con CONSER, ambos en colaboración con otros investigadores, incluyendo ACIRI. Ns siempre ha incluido contribuciones de otros investigadores, incluyendo el código para conexiones inalámbricas de los proyectos UCB Daedalus y CMU Monarca, y de Sun Microsystems. [13]

2.3.1 Requerimientos del Sistema

Ns está diseñado para ser instalado en Linux, y existen guías completas para hacerlo en Debian, Fedora y Ubuntu, aunque puede ser instalado en Windows 95 o versiones superiores utilizando el software Cygwin. [13]

Se recomienda instalar las versiones ns-allinone que incluyen otros programas necesarios para el funcionamiento del ns (Tcl, Tk, Otcl, TclCL, Nam, Xgraph).

Los requerimientos mínimos del sistema son [8]:

Procesador Pentium II de 200 MHz o equivalente.
32 MBs de RAM
320 MBs libres de espacio en el disco.

2.3.2 ¿Qué se puede hacer con la herramienta?

Ns es una herramienta con un gran número de aplicaciones, protocolos, modelos de tráfico, elementos de la red, tipos de red entre otras cosas.

En forma general ns puede crear [13]:

Redes terrestres, satelitales e inalámbricas; con diferentes tipos de algoritmos (DV, LS, PIM-DM, PIM-SM, AODV, DSR).

Fuentes de tráfico como web, ftp, telnet, cbr, http, tráfico estocástico, etc.

Fallas, incluye determinísticas, pérdidas probabilísticas, falla de conexiones, etc.

Varias disciplinas de cola (drop-tail, RED, FQ, SFQ, DRR, etc.) y QoS o calidad de servicio (ejemplo, InstServ y Diffserv).

En forma general Ns permite visualizar [13]:

Flujo de datos, construcción de colas y pérdidas de paquetes.

Comportamiento de protocolos: principio lento de TCP, self-clocking, control de congestión, retransmisión rápida y recuperación.

Movimiento de los nodos en conexiones inalámbricas.

Anotaciones para sobresaltar eventos importantes.

Estado del protocolo (ejemplo, TCP cwnd).

2.3.3 Interfaz de usuario

Para el diseño de cualquier red es necesario hacerlo por línea de código, y la visualización animada de los resultados debe hacerse por medio de la aplicación Network Animator, mejor conocido como NAM.

Ns también genera archivos de texto con información detallada de los eventos ocurridos durante la simulación.

Para observar los resultados en NAM es necesario incluir las siguientes líneas de código (en cursiva):

```
set ns [new Simulator]

#Archivo del tráfico generado y archivo para el NAM
ns trace-all [open uislan.out w]
$ns namtrace-all [open uislan.nam w]

#Definición del procedimiento "finalizar"
proc finalizar {} {
    global ns
    $ns flush-trace
    exec nam out.nam &
    exit 0
}
```

En [9] se encuentra la explicación detallada línea por línea del anterior código.

2.3.4 Resultados con ns2

Ns genera archivos de trazas en código ASCII, organizándolos en 12 columnas que son [9]:

Evento: puede tener 4 valores r (recibido), + (en cola), - (salida de cola), d (perdido).

Tiempo: en el cual ocurrió el evento.

Del nodo: el último nodo del cual salió el paquete.

Al nodo: el próximo nodo al que se dirige el paquete.

Tipo de paquete: CBR, TCP, etc.

Tamaño del paquete.

Banderas.

Id del flujo.

Dirección fuente (nodo fuente).

Dirección destino (nodo destino).

Número de secuencia del paquete.

Id del paquete: este número es único.

Las columnas vienen organizadas en ese orden.

De estos archivos se utilizan en la investigación las columnas 1, 2, 5, 9 y 10.

Además de crear estas tablas, se puede graficar tráfico contra tiempo con el software Xgraph que viene incluido al instalar ns2.

2.3.5 ¿Por qué ns2?

Es la primer herramienta de simulación de tráfico, con casi 20 años de experiencia y utilizada como modelo para otras herramientas de simulación, creada exclusivamente con fines investigativos.

Es completamente gratuito, trabaja sobre Linux, sus guías, manuales y laboratorios son también gratuitos, y es actualizada constantemente sin ningún costo.

Permite la manipulación total de cada componente de la red, sin estar sujeto a parámetros de propietario.

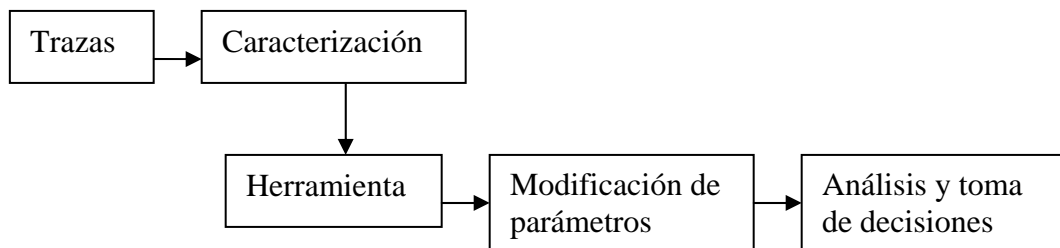
Ns tiene numerosas investigaciones importantes que soportan resultados de la herramienta, en especial para el estudio del fenómeno autosimilar.

No requiere la inclusión de nuevas librerías como en el caso de algunas herramientas.

2.4 Modelo de pruebas

Independientemente de la herramienta utilizada para generar tráfico artificial, se debe seguir un modelo de pruebas para la obtención de simulaciones más cercanas al sistema real, que permita experimentar cambios de la red a priori, evitando posibles inconvenientes que finalmente resultan en pérdidas monetarias por la mala utilización del recurso.

Figura 10 Modelo de pruebas.



La figura 10 muestra el modelo de pruebas para la investigación. Comenzando por la adquisición de trazas de tráfico, se caracteriza por medio de algún modelo probabilístico como Poisson, cadenas de Markov, Autosimilitud, esto con el fin de entender en forma matemática el comportamiento de la red, y de acuerdo con estos resultados se escoge una herramienta de simulación que pueda seguir la función de distribución de los datos reales y su forma específica de comportamiento.

La red simulada finalmente es el resultado más importante del modelo de pruebas, una vez obtenido, se podrán hacer modificaciones a los parámetros del sistema, que se traduce en la posibilidad de simular escenarios que intuyen las consecuencias generadas por cambios en la red, por medio de un análisis previo para la toma de decisiones.

3. AUTOSIMILITUD

La medida utilizada para determinar el grado de autosimilaridad en una red es el parámetro de Hurst, nombrado así por Harold Edwin Hurst.

Hurst estudió el Nilo y otros ríos con el fin de determinar los requerimientos del diseño para hacer represas de agua. Hurst contó con 800 años de registros de los niveles de agua del Nilo para diseñar una reserva que nunca se desbordara y que tampoco se secara. La clave para desarrollar dicho modelo estaba en medir la varianza del nivel del agua del río que alimentaba la reserva. Hurst descubrió que existían ríos con fluctuaciones pequeñas, y ríos con fluctuaciones de rangos amplios en diferentes escalas de tiempo. [4]

Numerosas investigaciones demuestran la existencia de las dinámicas fractales del comportamiento autosimilar en las redes LAN, MAN y WAN; por esta razón se dedicará un capítulo entero a la explicación de dicho fenómeno y su parámetro de medida H.

3.1 Concepto de Autosimilaridad [1, 2, 3, 4, 6, 22, 23]

La autosimilaridad es una característica en donde existe similaridad estructural en diferentes escalas en una dimensión específica, como el tiempo o el espacio. En otras palabras, existe un patrón que se repite a sí mismo al cambiar su escala de medida, y las estadísticas del proceso no varían. Esta característica puede no tener límite de escalas en modelos matemáticos aunque en la realidad se observa solamente para un rango de escalas definido. La figura 11 ilustra mejor la autosimilaridad de un fenómeno en el tiempo, que corresponde al volumen de tráfico en kilobytes para una interfaz del core switch de la Universidad Industrial de Santander. Dicha interfaz corresponde a la subred donde se ubican los servidores de interés para este trabajo.

Como se puede observar, al cambiar la escala temporal se siguen observando ráfagas cortas. Los modelos de tráfico convencionales no reflejan esta situación adecuadamente, ya que al cambiar la escala dichas ráfagas se suavizan. Como se observa en la figura 12, al agregar⁸ el tráfico a un segundo el modelo de Poisson suaviza la curva, mientras el modelo autosimilar sigue presentando ráfagas.

⁸ El término agregar se utiliza para decir que el tráfico total transcurrido durante una unidad de la escala de tiempo utilizada se suma, y corresponde a una medida de Kbytes por unidad de tiempo. La granularidad es que tan fino se hace el agregado de los datos.

Figura 11 Trazas de tráfico tomadas del core switch de la UIS en tres escalas de tiempo.

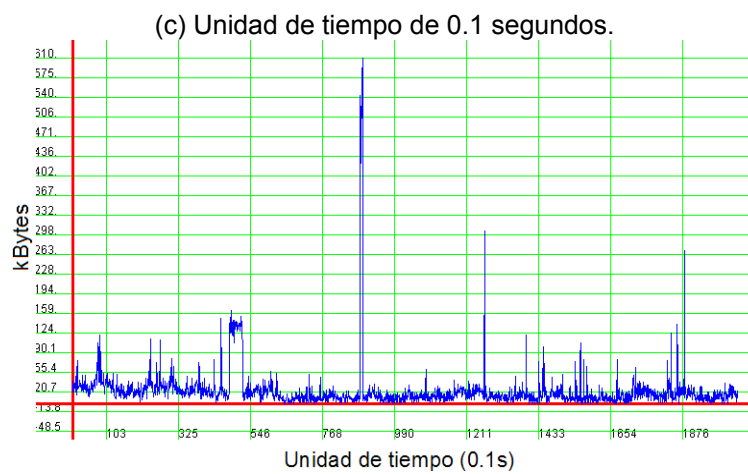
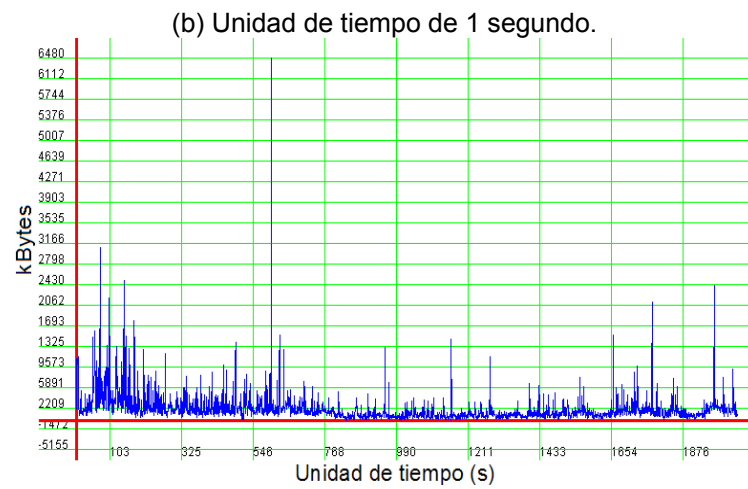
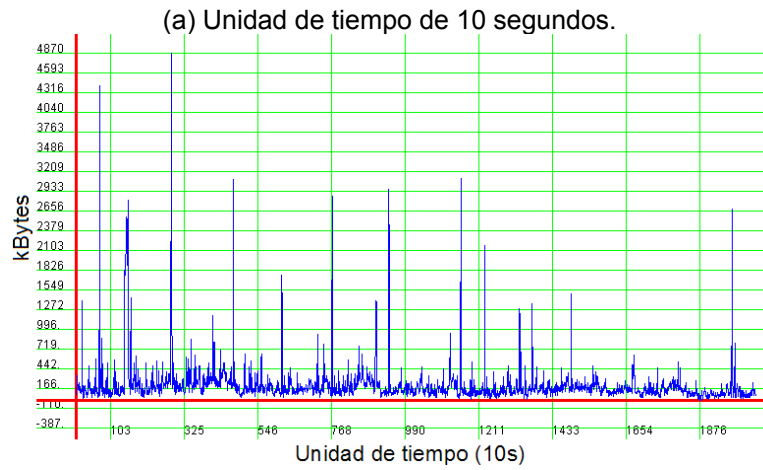
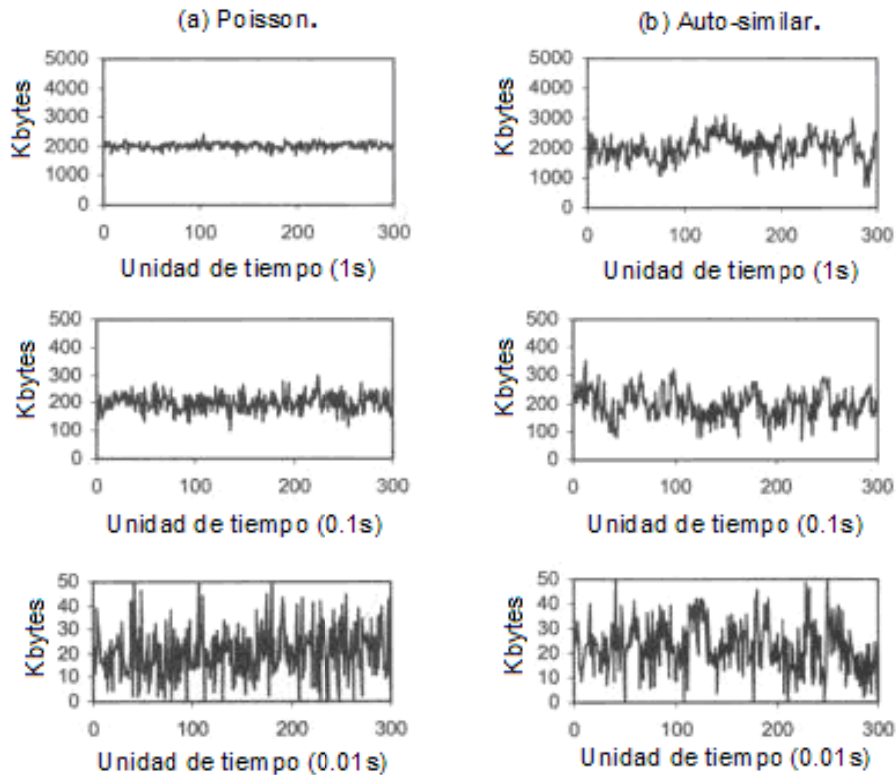


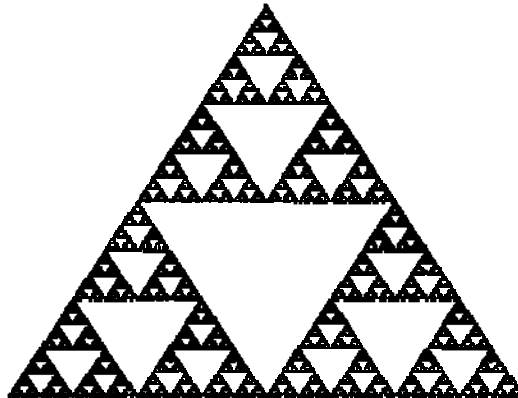
Figura 12 Poisson contra auto-similitud. [4]



La autosimilaridad se asocia con la geometría fractal, que describe objetos sin cambios de apariencia en diferentes escalas de medida. El concepto fractal incluye estadística y dinámica cuya descripción requiere espacios de dimensionalidad diversa. Algunos ejemplos de fractales son el triángulo de Sierpinski (geometría), ley de Pareto (economía), ley de Zipf (lingüística), ley de Lotka (sociología), ley de Willis (biología), el espectro cardíaco (medicina), mapas determinísticos (modelamiento telegráfico). La figura 13 muestra un ejemplo de fractal en la dimensión del espacio.

La conducta autosimilar está ligada también a la teoría del caos. Un sistema caótico es altamente sensible a las condiciones iniciales del sistema, de manera que cambios pequeños en las condiciones iniciales pueden modificar el comportamiento del sistema a largo plazo. Una de las características fundamentales de estos sistemas es que su aleatoriedad crea formas determinísticas, en consecuencia algunos fenómenos en la naturaleza aparentemente aleatorios tienen un orden subyacente. Ejemplos de la teoría del caos se encuentran en la ecología, predicciones de población biológica, el corazón humano, ciencias computacionales, imágenes fractales.

Figura 13 Triángulo de Sierpinski, utilizado en telecomunicaciones. [10]



3.2 El parámetro de Hurst

El parámetro de Hurst indica el grado de autosimilitud de una serie de datos y su valor oscila entre $0.5 \leq H \leq 1.0$. Cuando H tiende a 0.5 la autosimilaridad es baja, o ninguna en caso que sea igual a 0.5; cuando tiende a 1.0 es muy autosimilar, o exactamente autosimilar cuando es igual a 1.0.

Para conocer dicho parámetro en una red es necesario conformar una serie con los tiempos de llegada (X_n) de datagramas a través de diferentes escalas de tiempo. Un H cercano a 1.0 para el tráfico representa que un fragmento de dicha serie presentará un comportamiento muy similar a la serie original, mientras que un H cercano a 0.5 tanto el fragmento como la serie original no presentan ninguna similitud.

El parámetro de Hurst es la base para modelar una red. El estudio del tráfico se desarrolla en dos tipos diferentes de fuentes: los modelos que logran caracterizar fuentes individuales de tráfico y los modelos para fuentes de tráfico agregado. Independientemente del tipo de estudio que se realice, ambos presentan normalmente un valor de Hurst entre $0.7 < H < 0.9$ [1].

3.3 Formas de detectar la Autosimilaridad

Algunas de las formas más utilizadas para medir el grado de autosimilaridad en un sistema son:

- Dependencia a largo plazo
- Decaimiento lento de la varianza
- Rango reescalado o R/S
- Periodograma
- Estimador de Whittle
- Otros

Las figuras mostradas a continuación pertenecen a la misma serie de datos, mostrados en la figura 11, y se pueden utilizar para hacer comparaciones entre las mismas.

En esta investigación, se calculará el parámetro de Hurst (indicador de autosimilaridad) mediante el Periodograma y el estimador de Whittle, debido a su exactitud para calcular dicho parámetro.

3.3.1 Dependencia a largo plazo [1]

La dependencia a largo plazo o LRD (Long Range Dependence) refleja la persistencia del fenómeno autosimilar en una serie de datos, y es definida en términos del comportamiento de la función de autocorrelación.

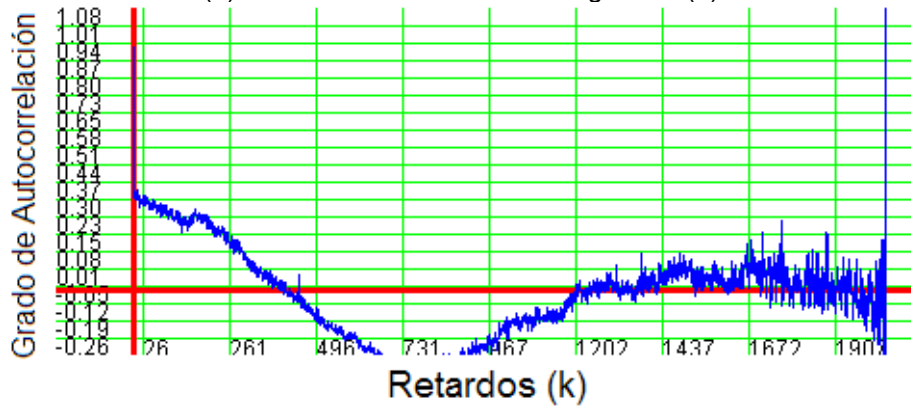
La autocorrelación es la medida estadística de la relación, si existe, entre la serie temporal discreta de datos (X_n) y sí misma, desfasada en el tiempo en un factor k (X_{n+k}). Dicho de otra forma, que tan parecidas son la serie de datos original y la desplazada. La función de autocorrelación oscila en el rango de uno y menos uno. Uno significa que la serie de datos original y la desplazada están correlacionadas, o son idénticas, al superponerlas; menos uno representa una anticorrelación al superponerlos, en otras palabras, son idénticas pero una es inversa con respecto a la otra; cero significa que no hay correlación, o similitud, entre las series de datos.

Para la mayoría de procesos tipo Poisson, la autocorrelación es nula. La figura 14 (a) muestra la autocorrelación para la serie de datos de la figura 11 (a), el eje x corresponde a cada uno de los k desplazamientos de la serie y el eje y la autocorrelación; la figura 14 (b) muestra la autocorrelación de una serie de datos con distribución normal.

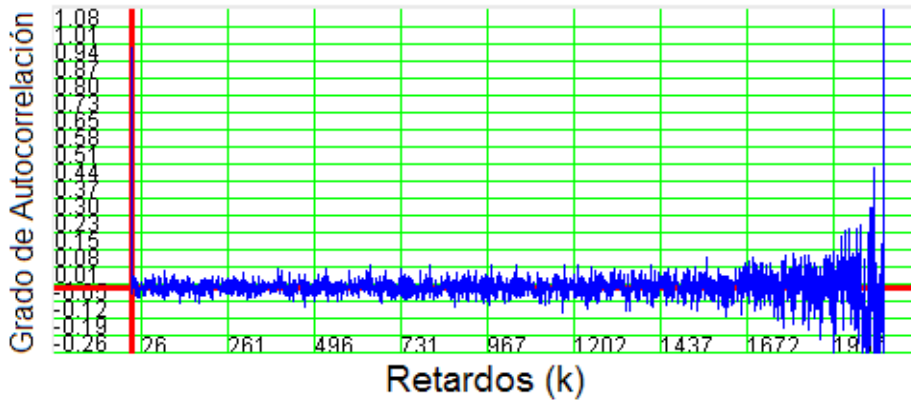
La autocorrelación solamente indica la existencia de autosimilaridad pero no su grado, por lo cual, es necesario utilizar otros métodos para determinarlo.

Figura 14 Función de autocorrelación.

(a) Para la serie de datos de la figura 11 (a).



(b) Para una serie de datos con distribución normal.



La función de autocorrelación se define mediante la ecuación 3.1. Las ecuaciones 3.2 y 3.3 corresponden a la función de varianza y la media en su orden.

$$R(k) = \frac{E[(X_n - \mu)(X_{n+k} - \mu)]}{\sigma^2} \quad (3.1)$$

Donde,

$$\sigma^2 = \text{Var}[X_n] = E[(X_n - \mu)^2] \quad (3.2)$$

$$\mu = E[X_n] \quad (3.3)$$

3.3.2 Decaimiento lento de la varianza [1]

La varianza es otro indicador de autosimilitud. Los procesos autosimilares presentan un decaimiento lento de la varianza al aumentar el tamaño del grano. En cambio, para la mayoría de procesos, como los de distribución normal, el decaimiento de la varianza es rápido con el aumento del tamaño del grano. Las ecuaciones 3.4 y 3.5 describen dicho comportamiento, con m como el tamaño del grano.

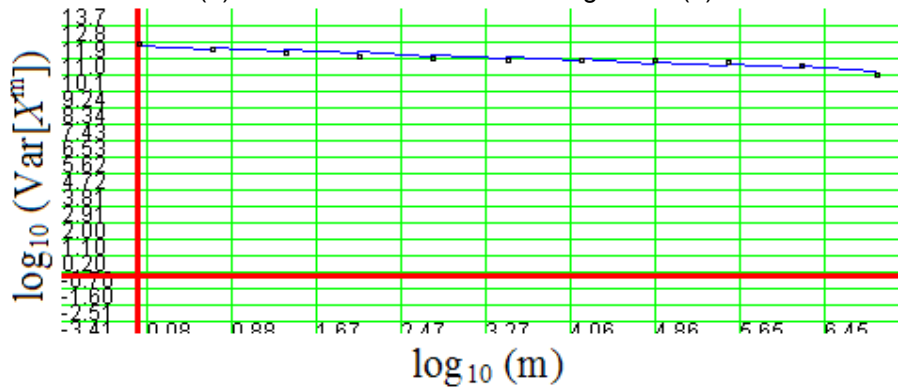
$$\text{Var}[X^m] = \frac{\text{VAR}[X]}{m^\beta} \quad (3.4)$$

Donde,

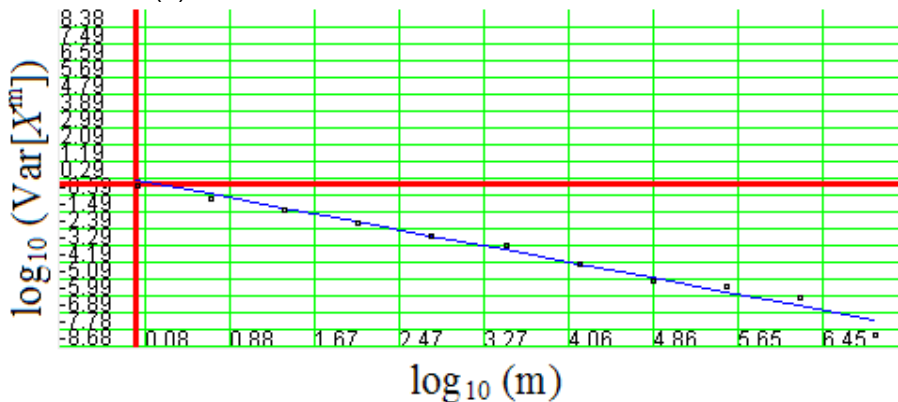
$$H = 1 - \frac{\beta}{2} \quad (3.5)$$

Figura 15 Decaimiento lento de la varianza.

(a) Para la serie de datos de la figura 11 (a).



(b) Para una serie de datos con distribución normal.



Una gráfica de varianza para un proceso normal tiene una pendiente de -1, un proceso autosimilar tiene una pendiente mayor. La figura 15 (a), muestra el comportamiento de la varianza para la serie de datos de la figura 11 (a), donde $m = 10$ segundos, cada dato de la serie representa todos los sucesos ocurridos durante ese intervalo de tiempo, y aunque el tamaño del grano es muy grande, se obtiene un $H = 0.905$. La figura 15 (b), presenta la gráfica para una serie de datos con distribución normal, donde $m = 1$ dato, y conduce a un $H = 0.462$. El cálculo de la pendiente de la figura 15 permite conocer el grado de autosimilaridad de los datos. Valores superiores de dicha pendiente corresponden a niveles más altos de autosimilitud.

3.3.3 Rango reescalado o R/S [7]

Hurst desarrolló la medida R/S adimensional, para caracterizar la variabilidad de una serie temporal de datos. Para una serie de datos $X(t, N)$, con una media \bar{X}_N , varianza S^2 , y rango R ; se obtiene la estadística $R(N)/S(N)$ en las ecuaciones 3.6 y 3.7.

$$R(N) = \max_{1 \leq t \leq N} [X(t, N)] - \min_{1 \leq t \leq N} [X(t, N)] \quad (3.6)$$

$$S(N) = \left(\frac{1}{N} \sum_{t=1}^N [x_t - \bar{x}_N]^2 \right)^{\frac{1}{2}} \quad (3.7)$$

Donde,

$$X(t, N) = \sum_{i=1}^k X_i - k\bar{X}_N \quad (3.8)$$

Para $k = 1, 2, \dots, n$.

Hurst encontró que los fenómenos naturales seguían la ley empírica mostrada en la ecuación 3.9.

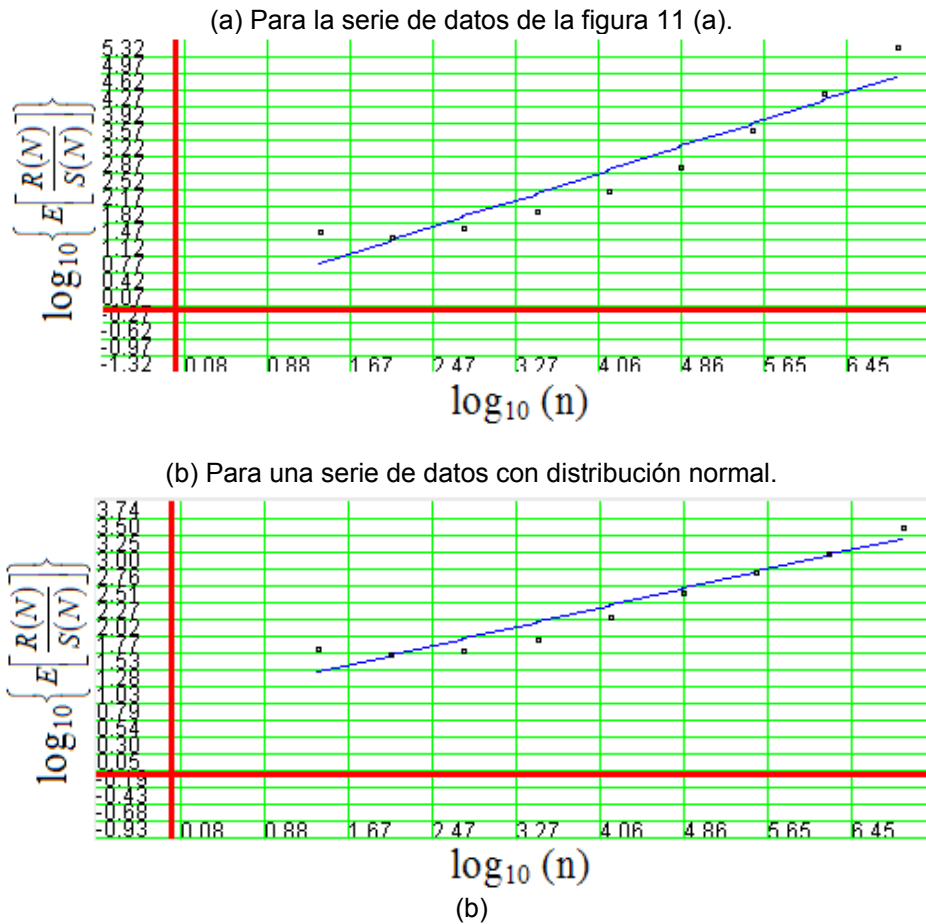
$$E \left[\frac{R(N)}{S(N)} \right] \sim cN^H \quad (3.9)$$

Si $N \rightarrow \infty$ y c es una constante finita independiente de N , aplicando logaritmos a ambos lados

$$\log \left\{ E \left[\frac{R(n)}{S(n)} \right] \right\} \sim H \log(n) + \log(c) \quad (3.10)$$

Entonces, se puede estimar H al graficar $\log \left\{ E \left[\frac{R(N)}{S(N)} \right] \right\}$ contra $\log(n)$, uniendo los puntos con una línea recta, su pendiente corresponde al parámetro H. La figura 16 (a) muestra una gráfica R/S para la serie de datos de la figura 11 (a), arrojando $H = 0.708$. La figura 16 (b) muestra el resultado R/S para una serie de datos con distribución normal, se obtiene un $H = 0.350$.

Figura 16 Rango Reescalado.



Es necesario indicar que el método R/S no es preciso, y da solo un estimado del valor H.

3.3.4 Periodograma [1, 2]

El periodograma es un método que ofrece un buen rigor estadístico. El periodograma (o función de intensidad) $I_N(\omega)$ es la densidad espectral estimada de un proceso estocástico discreto $X(t)$. El periodograma se calcula mediante la ecuación 3.11, donde X_k es la serie de datos, N es la longitud de X_k y ω es la frecuencia.

$$I_N(\omega) = \frac{1}{2\pi N} \left| \sum_{k=1}^N (X_k e^{jk\omega}) \right|^2 \quad (3.11)$$

Dado que la autosimilaridad afecta el comportamiento del espectro cuando ω tiende a 0, con lo cual la expresión en 3.11 se aproxima a $I_N(\omega) \sim |\omega|^{1-2H}$. Al graficar $\log[I_N(\omega)]$ versus $\log(\omega)$ (únicamente en frecuencias bajas), y agregando una línea recta a la curva, la pendiente se aproxima a $1-2H$.

En la figura 17 (a) se observan los resultados del periodograma para la serie de datos de la figura 11 (a), $H = 0.734$, indicando una autosimilaridad media. La figura 17 (b) muestra el resultado del periodograma para una serie de datos con distribución normal, se obtiene un $H = 0.350$, no hay autosimilitud.

3.3.5 Estimador de Whittle [1, 2]

El estimador de Whittle está basado en el periodograma y utiliza verosimilitud máxima. Se calcula mediante la ecuación 3.12 para $d = H - 0.5$.

$$Q(d) = \int_{-\pi}^{\pi} \frac{I_N(\omega)}{S(\omega)} \quad (3.12)$$

En la práctica la integral de la ecuación 3.12 es una sumatoria. Considerando la frecuencia del sistema $\omega_k = k \frac{2\pi}{N}$ para los valores de k desde 0 hasta $N-1$, se obtiene la ecuación 3.13.

$$Q(d) = \sum_{k=0}^{N-1} \frac{I_N(\omega)}{S(\omega)} \quad (3.13)$$

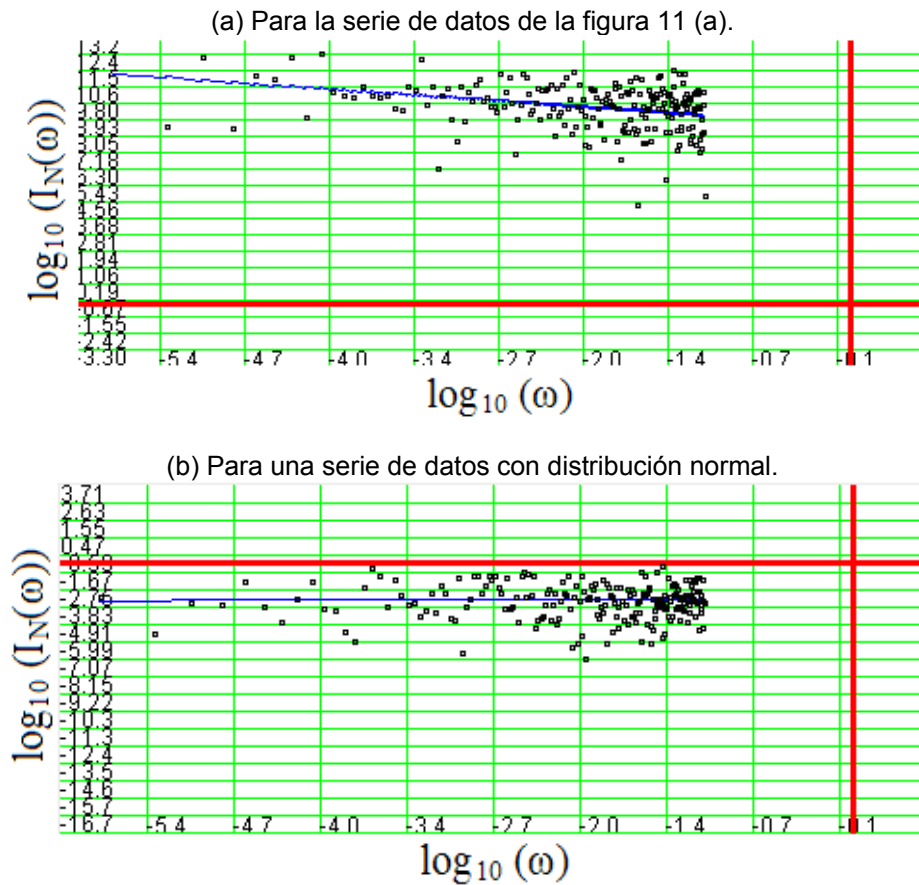
Donde la ecuación de densidad espectral se expresa mediante la ecuación 3.14, con una función que corresponde a $R(k)$.

$$S(\omega) = \sum_{k=-\infty}^{\infty} R(k) e^{-j2k\omega} \quad (3.14)$$

El estimador de Whittle tiene la limitante de ser paramétrico y por lo tanto requiere conocer la densidad espectral con anterioridad, lo cual se traduce en un alto costo computacional.

El H obtenido con el estimador de Whittle para los datos de la figura 11 (a) es de $H = 0.759$, para unos datos generados con distribución normal se encuentra que es de $H = 0.512$. Este estimador para el parámetro de Hurst es un método no gráfico.

Figura 17 Periodograma.



3.3.6 Otros

Otras formas para determinar la autosimilaridad de una serie de datos se pueden encontrar en [1, 2]. Los métodos que no se mencionan en este capítulo son adecuados para ciertas condiciones de tráfico, por eso no se explican en este documento.

4. RESULTADOS EN LA INTRANET UIS

La primera LAN de la UIS fue creada en 1988. La red no utilizaba ningún hub o switch, en su lugar las conexiones se realizaban con transceivers utilizando una ethernet en cable coaxial de 10Mbps para el enlace interno, desde luego no existía Internet en Colombia para la época. Posteriormente en 1995 se implementó un switch de 8 puertos con una velocidad de 10 Mbps en fibra óptica multimodo, al cual se conectaron hubs de 10 puertos con la misma velocidad del switch, y con estos se conectaban los equipos a la red desde los edificios. La UIS a su vez, adquiere acceso dedicado a la Internet con un ancho de banda de 128KBps.

En el año 2000 se adquirió el primer switch core con referencia Avaya P880 el cual utilizaba conexión 10/100/1000, además, para algunos edificios de la Universidad se implementaron switches Avaya P330 con conexión fast ethernet, que manejaban el tráfico de cada edificio haciendo la conexión al switch central. Algunos enlaces ya utilizaban protocolo Gigabitethernet, y la conexión dedicada a Internet creció a 512KBps.

Para el año 2006 el switch Avaya P880 fue reemplazado por un switch Extreme Networks BD 6808, el ancho de banda para el acceso dedicado a Internet ya había aumentado a 20Mbps, y en el 2008 los switches Avaya P330 fueron reemplazados por switches 3COM 4500G. Hasta la fecha la UIS cuenta con un acceso a Internet con 57Mbps de ancho de banda, y en su backBone utiliza pocos enlaces fast ethernet, en su mayoría son Gigabitethernet.

Con los años venideros la Intranet de la UIS va a cambiar y cada vez más rápido debido a la demanda de tráfico generada por sus usuarios. Realizar un estudio para mejorar el desempeño de la misma es necesario. Esta investigación permite entender como se comporta la red, y a futuro ayudará a tomar mejores decisiones cuando se planeen estos cambios.

4.1 Selección de las características del tráfico

Se seleccionaron los servidores de la DSI⁹ de la UIS que se encuentran en el mismo segmento y el tráfico de estos llega al mismo puerto del core switch institucional. Además, se seleccionaron por utilizar diferentes protocolos importantes como FTP, http, SMTP y DNS. Con esto se obtiene una porción de la

⁹ División de servicios de información.

red limitada en tamaño, pero que facilita la experimentación con datos reales de los protocolos de mayor uso en la mayoría de redes.

La figura 18 muestra el esquema utilizado para la toma de datos sobre la red, en el cual se muestreó el tráfico entrante y saliente (se aclara que los servidores son fuentes cuando transmiten tráfico y sumideros cuando reciben tráfico) del puerto asociado en el core switch al segmento de red seleccionado. Por supuesto en este puerto se registra tráfico de otras fuentes, que debe ser filtrada. Ver anexo A.

La figura 19 muestra la topología de la red actual. La red se encuentra separada por edificios, donde se encuentran los Switch 3COM 4500G. Cada edificio se conecta al core switch, y a éste, también se conecta el servidor WebSense (servidor para el filtro de contenidos), pasa por un media converter, después es analizado por el firewall, finalmente llega al administrador (packeteer) que regula el ancho de banda por aplicación, donde la información finalmente va rumbo a la Internet por los convenios que la Universidad tiene con las empresas que ofrecen el servicio. Las referencias de los dispositivos y los detalles de conexión y anchos de banda se pueden observar en la figura 19.

La intranet de la UIS maneja protocolo Gigaethernet., los retardos entre (p1) y (p2) son del orden de 1 ms, y entre (p1) y (p3) son de 2 ms (ver figura 19).

Figura 18 Esquema de evaluación.

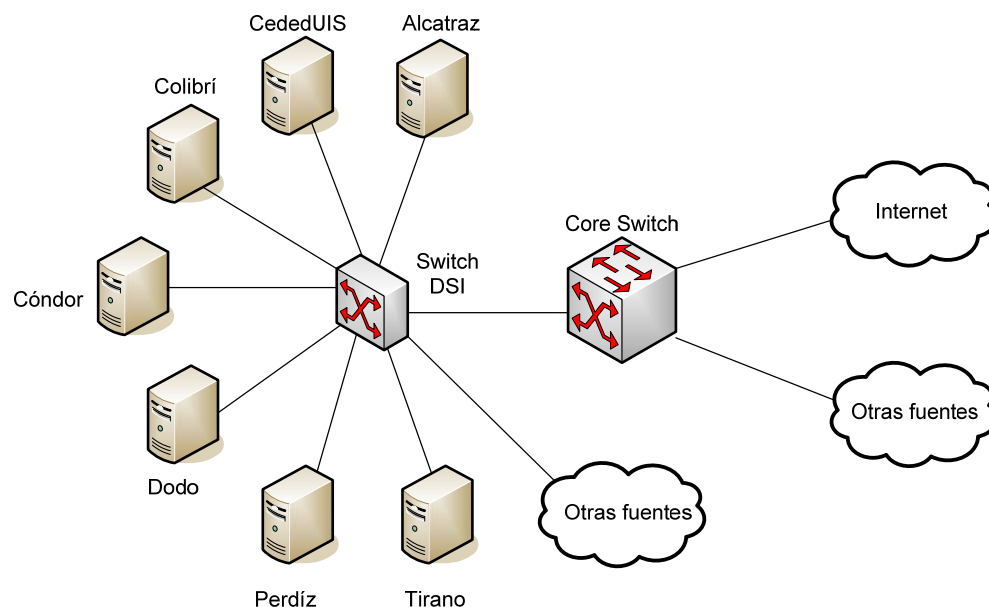
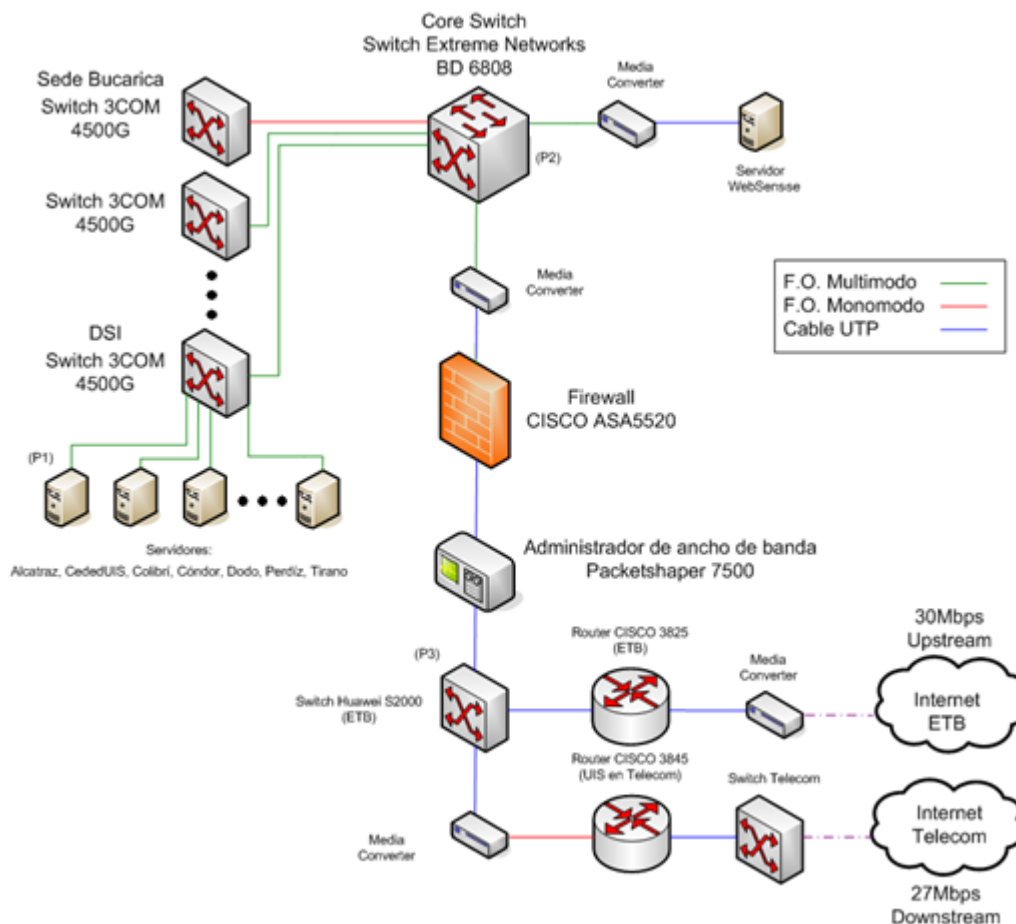


Figura 19 Topología de la Intranet de la Universidad Industrial de Santander.



4.2 Pruebas sobre la red

Sobre el Core Switch se realizó un registro del tráfico mediante el uso del software Wireshark (ver anexo A), el cual contiene los datagramas entrantes y salientes para cinco días de los servidores Alcatraz, CededUIS, Colibrí, Cándor, Dodo, Perdiz y Tirano.

Para el desarrollo del análisis de los registros, es necesario primero separar los datos de entrada y los datos de salida, y posteriormente totalizar el volumen de datos entrantes o salientes en una unidad de tiempo, la cual varía para realizar dos tipos de pruebas (las pruebas se explican a continuación, y se muestra un resumen en la tabla 2). Para ello se suman las longitudes de cada uno de los datagramas registrados en dicha unidad de tiempo. Con este resultado se obtiene una gráfica de volumen de tráfico contra tiempo.

Tabla 2 Resumen de pruebas realizadas al Core Switch de la UIS.

Prueba 1				Finalidad
Tamaño grano	Punto de registro	Longitud vector	Duración registro	Relación entre segmentos de tráfico, con cambio de escala en tiempo
10s	Core	2048	20480 s	
1s	Core	2048	2048 s	
0,1s	Core	2048	204,8 s	
Prueba 2				Finalidad
Tamaño grano	Punto de registro	Longitud vector	Duración registro	Relación de resultados con respecto al tamaño del grano.
10s	Core	2048	20480 s	
	Servidores	2048	20480 s	
0,625s	Core	32768	20480 s	
	Servidores	32768	20480 s	

La primera de estas pruebas consiste en comparar el comportamiento del tráfico agregado contra un segmento del mismo, para explorar la hipótesis de autosimilitud, según la cual deben presentar un comportamiento similar, como se propone en el capítulo 3. El tamaño para los granos utilizados en esta prueba es de 10 segundos para el registro de un día, 1 segundo que corresponde a un décimo del registro del día, y 0.1 segundos para una centésima del registro del día. La longitud del vector resultante para cada tipo de grano se ajusta para que sea el mismo.

La segunda consiste en comparar un registro consigo mismo para tamaños de grano diferentes y observar sus efectos en los resultados. El tamaño de los granos utilizados es de 10 segundos y 0.625 segundos. Por supuesto en esta prueba la longitud de los vectores es diferente. Para este tipo de prueba también se separa individualmente el tráfico de cada servidor para comparar las características de tráfico de cada uno contra los resultados del core switch.

El tiempo para obtener el registro de dichos datos por día es de 5,689 horas, o lo que equivale a 20480 segundos. Como el software [19] para calcular el parámetro de Hurst solamente trabaja con potencias de 2 se arreglan los vectores de la siguiente manera:

Se puede observar que al dividir el tiempo del registro en granos de 10 segundos el vector resultante será de una longitud de 2048. También, que al dividir el tiempo total del día en 10 se pueden obtener 10 vectores con longitud 2048 con grano de 1 segundo. Debido a la complejidad para calcular las características del tráfico, solo se toma un vector con grano de 0.1 segundos por cada vector de 1 segundo, a manera de muestra y observar si las características de tráfico se mantienen con el cambio del tamaño de grano.

4.3 Autosimilitud de la red

A continuación se muestran (prueba 1: tablas 3 a 7, figuras 20 a 24; prueba 2: tablas 8 a 13, figuras 25 a 30) los resultados del periodograma y el estimador de Whittle para determinar la autosimilitud en el core switch y cada uno de los servidores involucrados, de acuerdo con las pruebas mencionadas anteriormente.

Se separan las tablas y las figuras en (a) y (b), donde (a) son los resultados para los datos recibidos y (b) los resultados para los datos transmitidos, P y W hacen referencia a periodograma y estimador de Whittle respectivamente. Recuerde que un resultado de 0.5 en el parámetro de Hurst significa que no existe autosimilitud, y que un resultado de 1.0 que es completamente autosimilar, normalmente este parámetro oscila entre $0.7 < H < 0.9$. En las tablas, kB son los kiloBytes transmitidos en ese intervalo de tiempo.

Las tablas 3 a 7 presentan los resultados para la prueba 1. D0# hace referencia a los resultados para un grano de 10 segundos, H# para el grano de 1 segundo, y M# para el grano de 0.1 segundos.

Las tablas 8 a 13 presentan los resultados para la prueba 2. En las tablas, 0.625s y 10s hacen referencia a los tamaños de grano.

En las figuras el eje y corresponde al valor del parámetro de Hurst (H).

Tabla 3 Resultados prueba 1, para el core switch día 1.

(a) Recepción.											
H#	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	D01
kB	238780	133660	127140	152300	138510	71480	85777	84147	130550	175370	1352844
P	0.630	0.526	0.681	0.485	0.526	0.577	0.534	0.511	0.510	0.555	0.571
W	0.802	0.581	0.667	0.669	0.644	0.813	0.747	0.693	0.854	0.662	0.573
M#	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	
KB	16270	20855	11535	22617	20154	5140	5799	7049	9170	14323	
P	1.350	0.801	0.907	0.954	0.724	0.673	0.694	0.762	0.663	0.756	
W	0.999	0.999	0.881	0.999	0.999	0.998	0.667	0.851	0.807	0.607	
(b) Transmisión.											
H#	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	D01
kB	602050	543170	540820	520440	331760	255010	293020	408310	413190	542270	4466569
P	0.532	0.600	0.729	0.595	0.665	0.573	0.749	0.789	0.592	0.587	0.651
W	0.792	0.753	0.727	0.640	0.756	0.769	0.750	0.722	0.742	0.713	0.659
M#	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	
KBs	49436	52437	72722	40893	42223	24149	23166	38331	46410	84769	
P	0.877	0.559	0.547	0.694	0.671	0.761	0.586	0.678	0.744	0.984	
W	0.841	0.666	0.930	0.623	0.965	0.998	0.998	0.794	0.870	0.997	

Figura 20 Resultados prueba 1, para el core switch día 1.

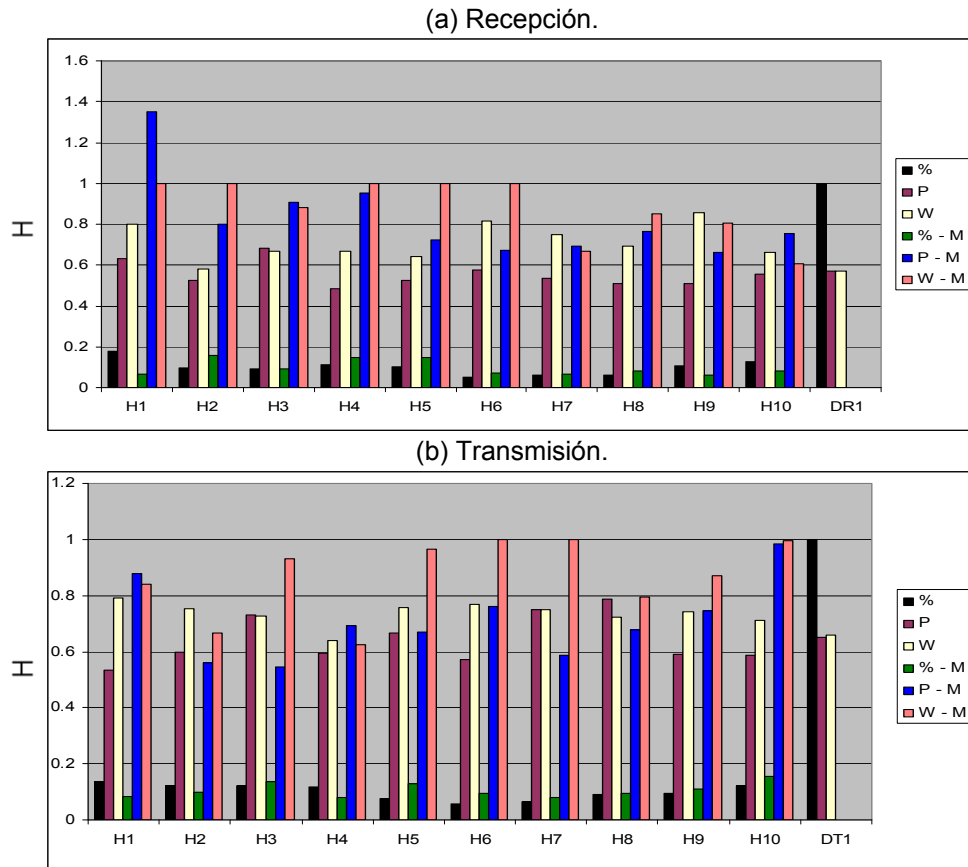


Tabla 4 Resultados prueba 1, para el core switch día 2.

(a) Recepción.											
H#	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	D02
kB	129180	155020	158480	113890	63916	50556	62871	114010	120150	141630	1129100
P	0.541	0.474	0.517	0.500	0.522	0.979	0.512	0.519	0.965	0.475	0.520
W	0.798	0.671	0.681	0.834	0.639	0.824	0.635	0.843	0.580	0.681	0.567
M#	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	
KB	11261	14431	15568	9758	5403	4293	11489	11432	23260	17822	
P	0.597	0.503	0.550	0.580	0.786	0.961	1.340	0.513	0.555	0.576	
W	0.634	0.980	0.794	0.981	0.835	0.845	0.999	0.847	0.998	0.998	
(b) Transmisión.											
H#	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	D02
kB	571850	612310	529560	406340	272360	164090	200730	345200	394280	445950	3947500
P	0.693	0.554	0.572	0.603	0.553	0.647	0.803	0.795	0.795	0.625	0.687
W	0.830	0.778	0.708	0.659	0.812	0.719	0.709	0.672	0.672	0.807	0.720
M#	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	
KB	61602	69702	43106	36812	20098	13480	19478	38784	50296	43409	
P	1.181	0.711	0.744	0.551	0.495	0.979	0.520	0.541	0.526	0.695	
W	0.884	0.995	0.896	0.997	0.640	0.824	0.901	0.877	0.863	0.910	

Figura 21 Resultados prueba 1, para el core switch día 2.

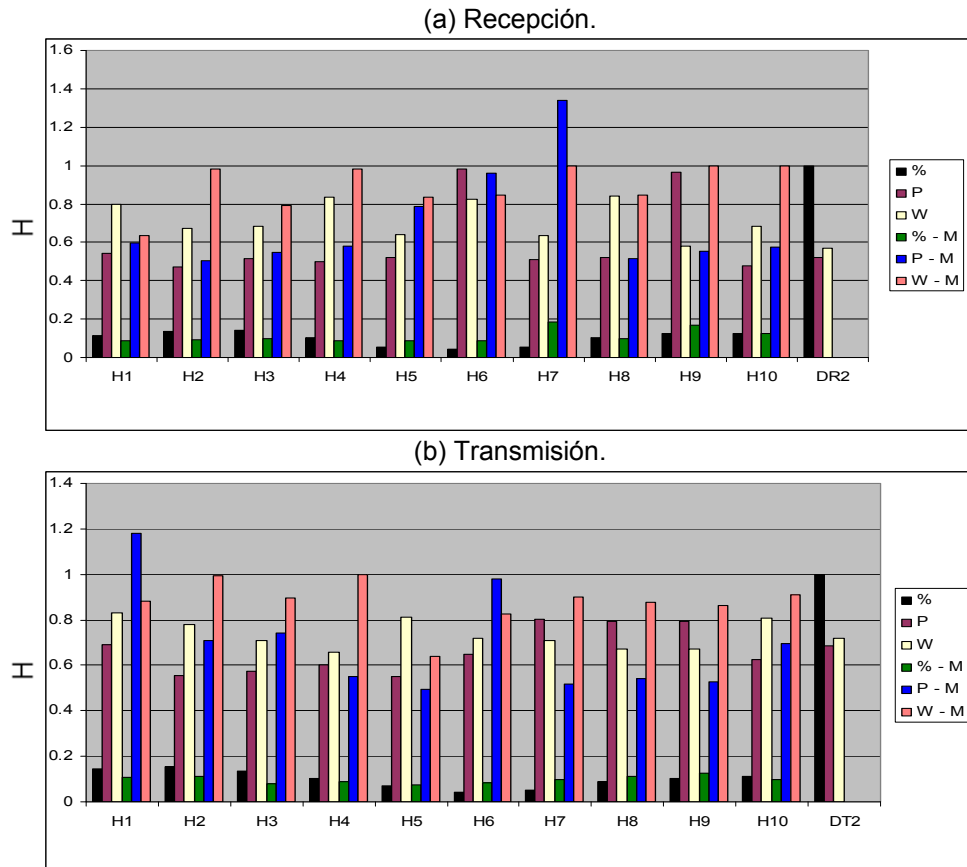


Tabla 5 Resultados prueba 1, para el core switch día 3.

(a) Recepción.											
H#	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	D03
kB	170070	138280	149620	87149	57307	97321	64157	113560	176060	138180	1209000
P	0.611	0.634	1.011	0.533	0.634	0.908	0.501	0.505	0.692	0.554	0.503
W	0.744	0.630	0.791	0.683	0.702	0.999	0.702	0.704	0.997	0.897	0.635
M#	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	
KB	30214	19091	10390	15104	5133	4087	4660	7851	40531	10941	
P	0.767	1.088	0.879	1.326	0.883	0.693	0.515	0.872	1.467	0.601	
W	0.998	0.970	0.876	0.999	0.842	0.666	0.787	0.924	0.999	0.998	
(b) Transmisión.											
H#	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	D03
kB	940810	509930	599860	329970	221050	221580	246060	411800	488130	426330	4396200
P	0.770	0.482	0.572	0.615	0.853	0.533	0.609	0.572	0.654	0.799	0.704
W	0.671	0.638	0.973	0.697	0.685	0.623	0.713	0.648	0.901	0.892	0.663
M#	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	
KB	93204	44208	53053	38976	39764	29692	13972	28026	41239	49159	
P	0.710	0.791	0.803	0.592	0.825	0.983	0.688	0.593	0.597	0.893	
W	0.997	0.804	0.993	0.888	0.991	0.999	0.711	0.805	0.832	0.999	

Figura 22 Resultados prueba 1, para el core switch día 3.

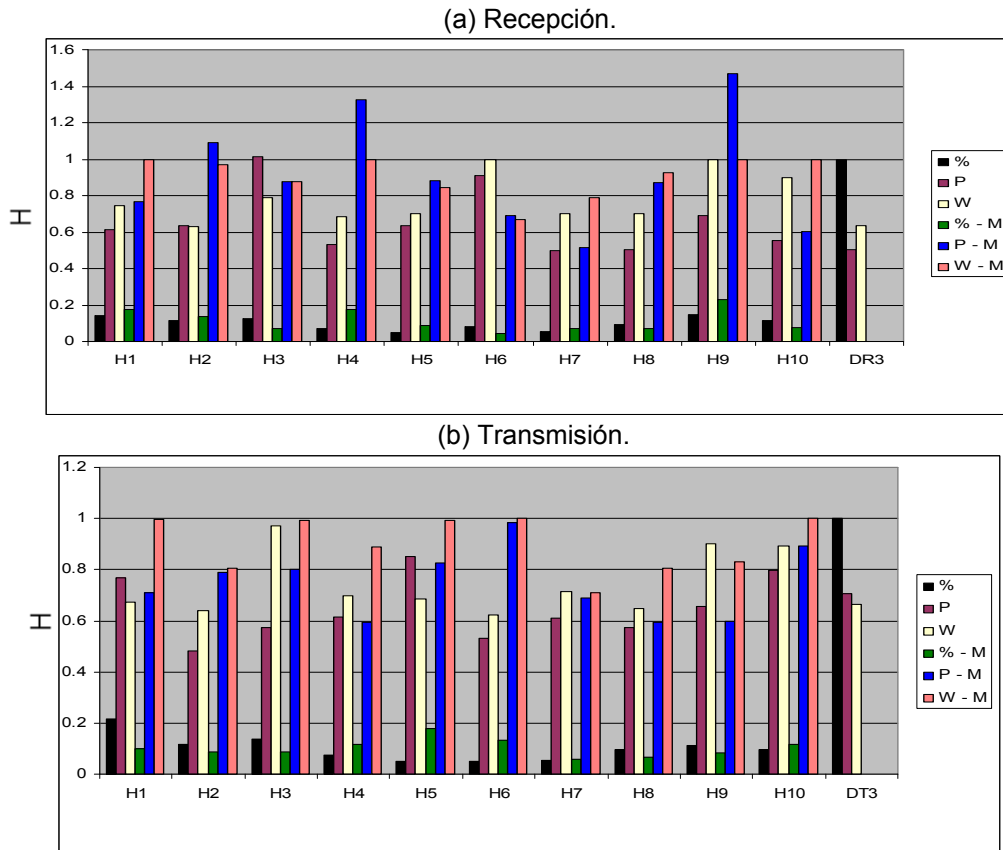


Tabla 6 Resultados prueba 1, para el core switch día 4.

(a) Recepción.											
H#	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	D04
kB	161900	102630	117380	71483	52758	60219	83823	109810	151740	140730	1070800
P	0.489	0.571	0.613	0.490	0.548	0.500	0.550	0.524	0.656	0.527	0.572
W	0.744	0.541	0.602	0.503	0.549	0.553	0.990	0.731	0.713	0.880	0.598
M#	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	
kB	8945	8943	9732	6919	3724	3804	7356	10779	10359	9443	
P	0.642	0.827	0.529	0.789	0.790	0.738	0.486	0.539	0.547	0.595	
W	0.820	0.879	0.654	0.785	0.823	0.788	0.638	0.733	0.997	0.712	
(b) Transmisión.											
H#	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	D04
KBs	455950	455970	458240	256880	232520	193300	302100	363700	373870	451300	3556000
P	0.688	0.659	0.589	0.806	0.754	0.619	0.540	0.579	0.656	0.527	0.628
W	0.631	0.590	0.663	0.695	0.765	0.781	0.639	0.706	0.713	0.880	0.643
M#	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	
kB	77055	27988	33113	25029	19233	12778	28057	61684	39645	59130	
P	0.614	0.711	0.600	0.745	0.645	0.731	0.850	0.812	0.689	0.980	
W	0.984	0.808	0.689	0.713	0.830	0.734	0.650	0.997	0.950	0.999	

Figura 23 Resultados prueba 1, para el core switch día 4.

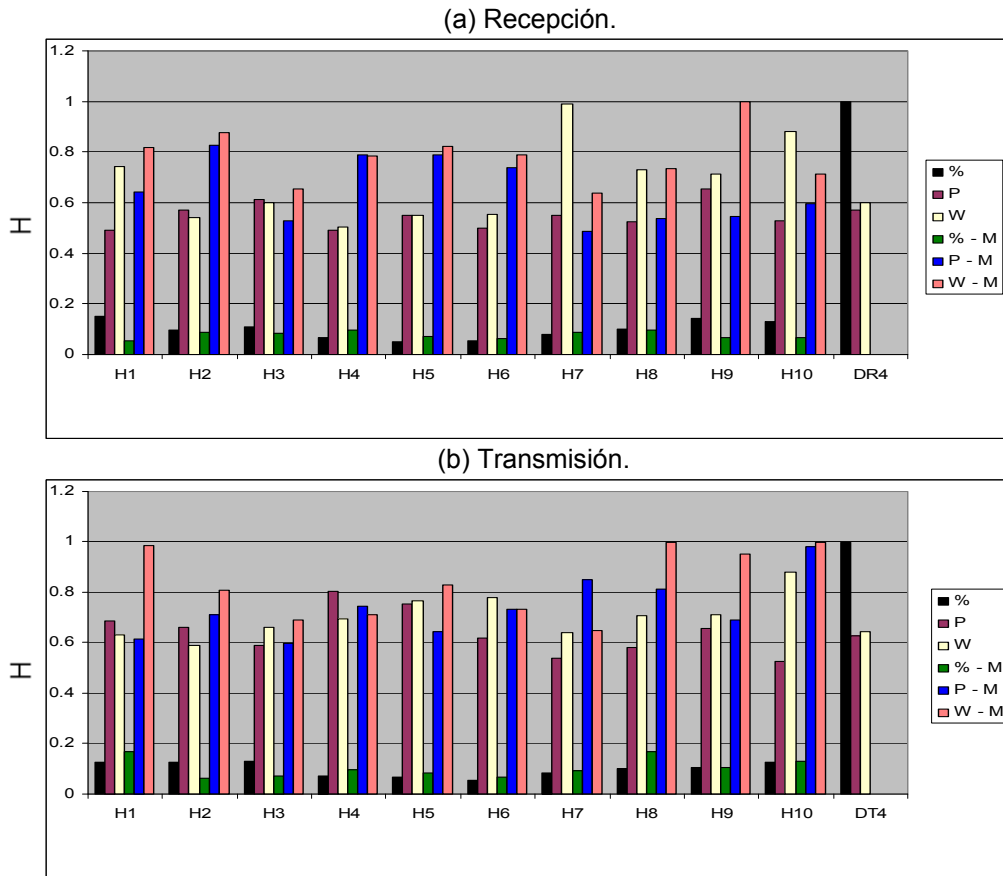


Tabla 7 Resultados prueba 1, para el core switch día 5.

(a) Recepción.											
H#	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	D05
kB	169280	146980	95398	78142	84785	67859	73922	118410	107330	82619	1039900
P	0.474	0.569	0.613	0.909	0.565	0.592	0.536	0.545	0.553	0.619	0.694
W	0.937	0.996	0.572	0.924	0.712	0.691	0.681	0.822	0.997	0.594	0.620
M#	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	
kB	25483	10094	11244	8407	10448	5723	5753	34912	8403	8162	
P	1.264	0.667	0.530	0.689	0.632	0.781	0.786	1.550	0.861	0.887	
W	0.999	0.826	0.915	0.708	0.998	0.911	0.893	0.997	0.831	0.890	
(b) Transmisión.											
H#	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	D05
kB	515360	377890	472580	237840	236450	201740	262760	293290	370560	248990	3227700
P	0.589	0.694	0.648	0.464	0.651	0.576	0.520	0.603	0.942	0.687	0.747
W	0.738	0.738	0.685	0.816	0.624	0.621	0.808	0.664	0.713	0.768	0.678
M#	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	
kB	128720	30683	64219	31921	42674	14160	18173	33356	84402	26428	
P	1.166	0.691	0.606	0.950	0.566	0.768	0.789	0.651	0.926	0.677	
W	0.999	0.677	0.983	0.817	0.974	0.694	0.738	0.706	0.951	0.672	

Figura 24 Resultados prueba 1, para el core switch día 5.

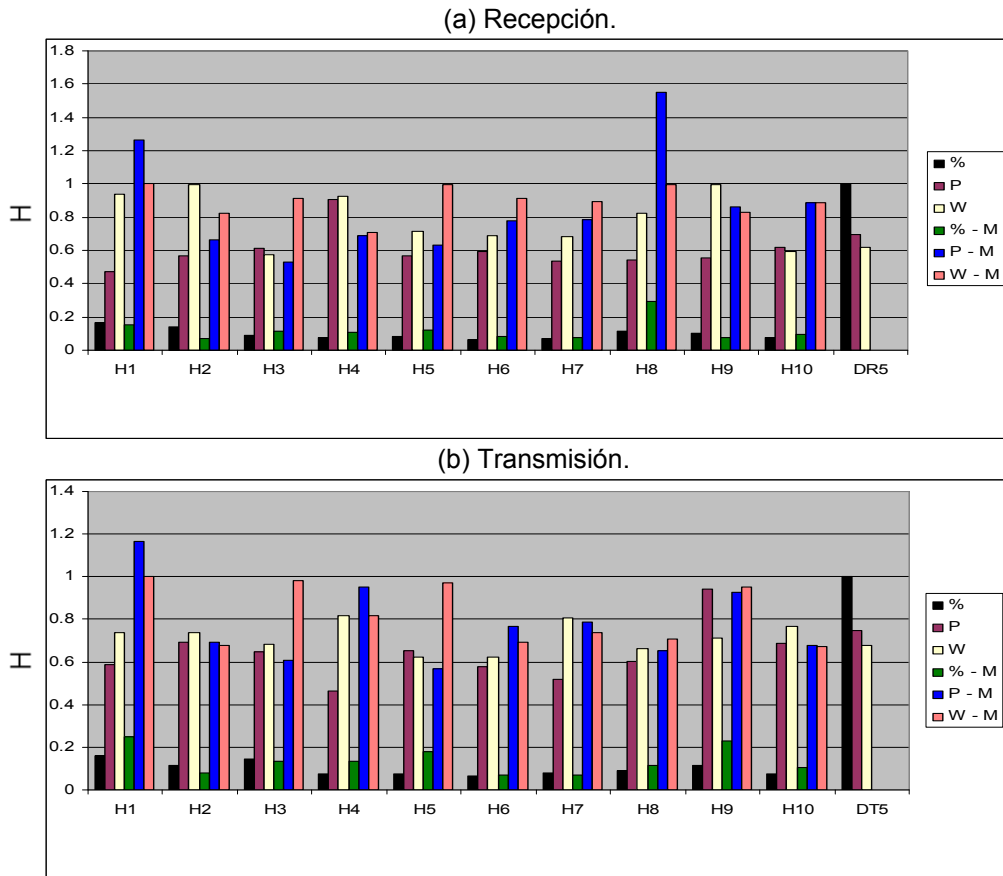


Tabla 8 Resultados prueba 2, para el core switch día 1.

(a) Recepción.								
0.625s	Condor	Dodo	Tirano	CededUIS	Perdiz	Colibri	Alcatraz	Core
kB	635000	677000	21539	3124	1454	5578	9150	1352845
P	0.588	0.569	0.694	0.739	0.866	0.622	0.786	0.588
W	0.836	0.935	0.721	0.566	0.731	0.647	0.817	0.842
10s	Condor	Dodo	Tirano	CededUIS	Perdiz	Colibri	Alcatraz	Core
P	0.549	0.533	0.835	0.706	0.669	0.662	0.713	0.571
W	0.545	0.558	0.798	0.644	0.745	0.694	0.927	0.573
%	0.4694	0.5004	0.0159	0.0023	0.0011	0.0041	0.0068	1.0000
(b) Transmisión.								
0.625s	Condor	Dodo	Tirano	CededUIS	Perdiz	Colibri	Alcatraz	Core
kB	1650000	2730000	15383	18264	12673	13107	27142	4466569
P	0.615	0.684	0.659	0.694	1.227	0.604	0.798	0.628
W	0.830	0.800	0.693	0.771	0.999	0.581	0.625	0.828
10s	Condor	Dodo	Tirano	CededUIS	Perdiz	Colibri	Alcatraz	Core
P	0.548	0.644	0.847	0.757	0.710	0.671	0.911	0.651
W	0.612	0.743	0.781	0.691	0.745	0.691	0.968	0.659
%	0.3694	0.6112	0.0034	0.0041	0.0028	0.0029	0.0061	1.0000

Figura 25 Resultados prueba 2, para el core switch día 1.

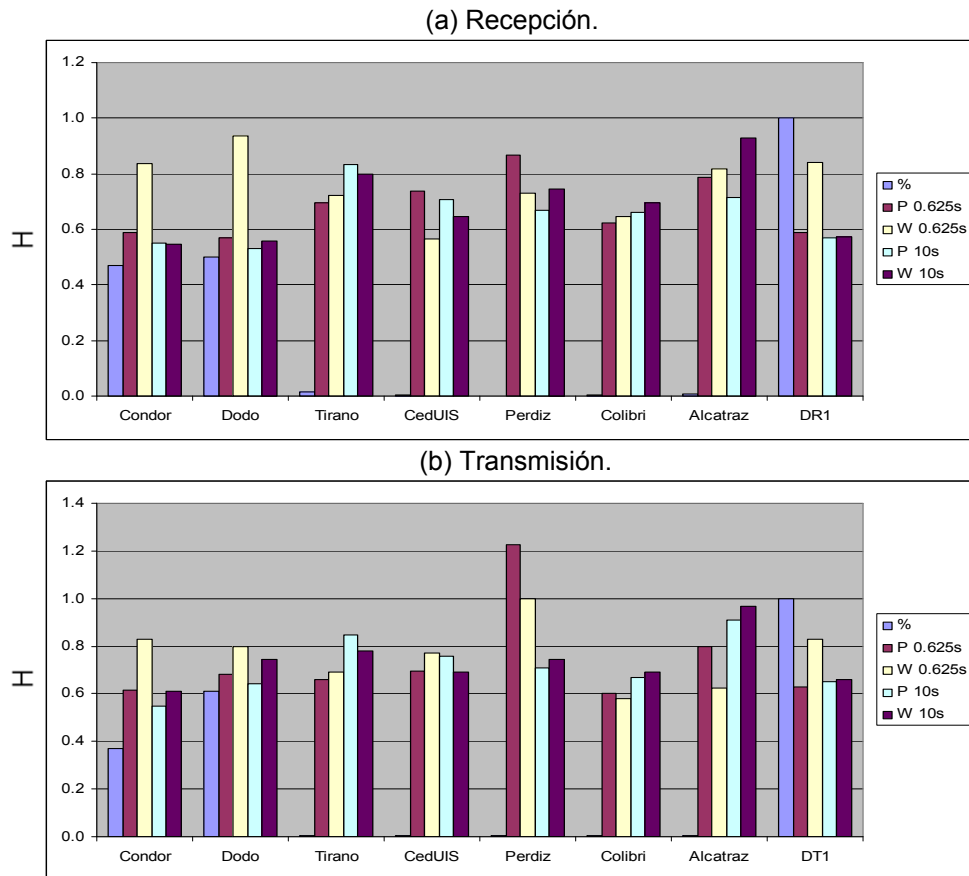


Tabla 9 Resultados prueba 2, para el core switch día 2.

(a) Recepción.								
0.625s	Condor	Dodo	Tirano	CededUIS	Perdiz	Colibri	Alcatraz	Core
kB	280310	817040	16693	2852	561	1494	10147	1129100
P	0.545	0.558	0.703	0.820	0.700	0.613	0.825	0.543
W	0.731	0.999	0.724	0.896	0.643	0.649	0.812	0.801
10s	Condor	Dodo	Tirano	CededUIS	Perdiz	Colibri	Alcatraz	Core
P	0.513	0.567	0.937	0.849	0.536	0.639	0.715	0.520
W	0.516	0.585	0.812	0.781	0.826	0.687	0.728	0.567
%	0.2483	0.7236	0.0148	0.0025	0.0005	0.0013	0.0090	1.0000
(b) Transmisión.								
0.625s	Condor	Dodo	Tirano	CededUIS	Perdiz	Colibri	Alcatraz	Core
kB	1508300	2366200	11757	27346	1423	3890	28621	3947500
P	0.651	0.567	0.680	1.036	0.647	0.638	0.703	0.666
W	0.743	0.585	0.693	0.969	0.650	0.568	0.646	0.755
10s	Condor	Dodo	Tirano	CededUIS	Perdiz	Colibri	Alcatraz	Core
P	0.713	0.734	0.890	1.059	0.493	0.602	0.715	0.687
W	0.683	0.759	0.793	0.997	0.681	0.650	0.742	0.720
%	0.3821	0.5994	0.0030	0.0069	0.0004	0.0010	0.0073	1.0000

Figura 26 Resultados prueba 2, para el core switch día 2.

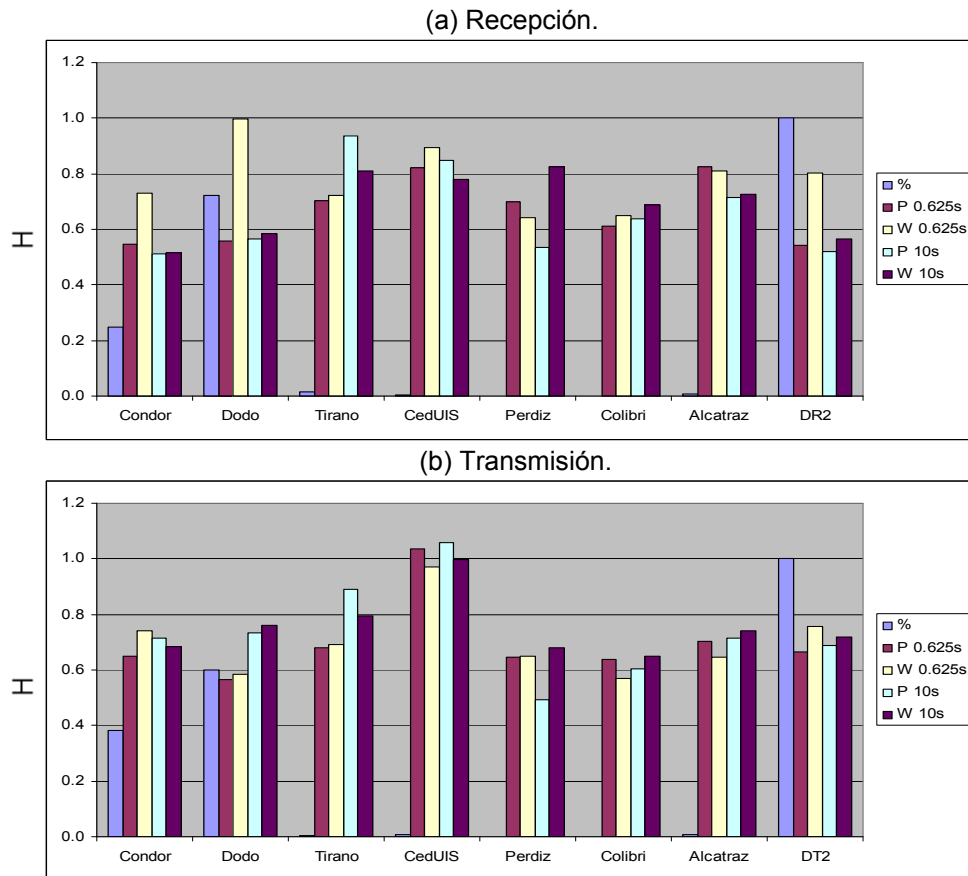


Tabla 10 Resultados prueba 2, para el core switch día 3.

(a) Recepción.								
0.625s	Condor	Dodo	Tirano	CededUIS	Perdiz	Colibri	Alcatraz	Core
kB	524120	642600	19524	4530	1172	6084	10948	1209000
P	0.749	0.553	0.665	0.959	0.719	0.701	0.889	0.728
W	0.997	0.889	0.722	0.965	0.730	0.702	0.999	0.987
10 s	Condor	Dodo	Tirano	CededUIS	Perdiz	Colibri	Alcatraz	Core
P	0.534	0.536	0.832	0.571	0.613	0.605	0.540	0.503
W	0.634	0.587	0.802	0.708	0.713	0.647	0.970	0.635
%	0.4335	0.5315	0.0161	0.0037	0.0010	0.0050	0.0091	1.0000
(b) Transmisión.								
0.625s	Condor	Dodo	Tirano	CededUIS	Perdiz	Colibri	Alcatraz	Core
kB	2141800	2172000	15285	33017	3288	12260	18519	4396200
P	0.666	0.654	0.679	0.596	0.540	0.606	0.783	0.660
W	0.773	0.624	0.690	0.549	0.563	0.557	0.649	0.750
10s	Condor	Dodo	Tirano	CededUIS	Perdiz	Colibri	Alcatraz	Core
P	0.710	0.632	0.827	0.447	0.635	0.573	0.588	0.704
W	0.648	0.660	0.795	0.709	0.583	0.628	0.862	0.663
%	0.4872	0.4941	0.0035	0.0075	0.0007	0.0028	0.0042	1.0000

Figura 27 Resultados prueba 2, para el core switch día 3.

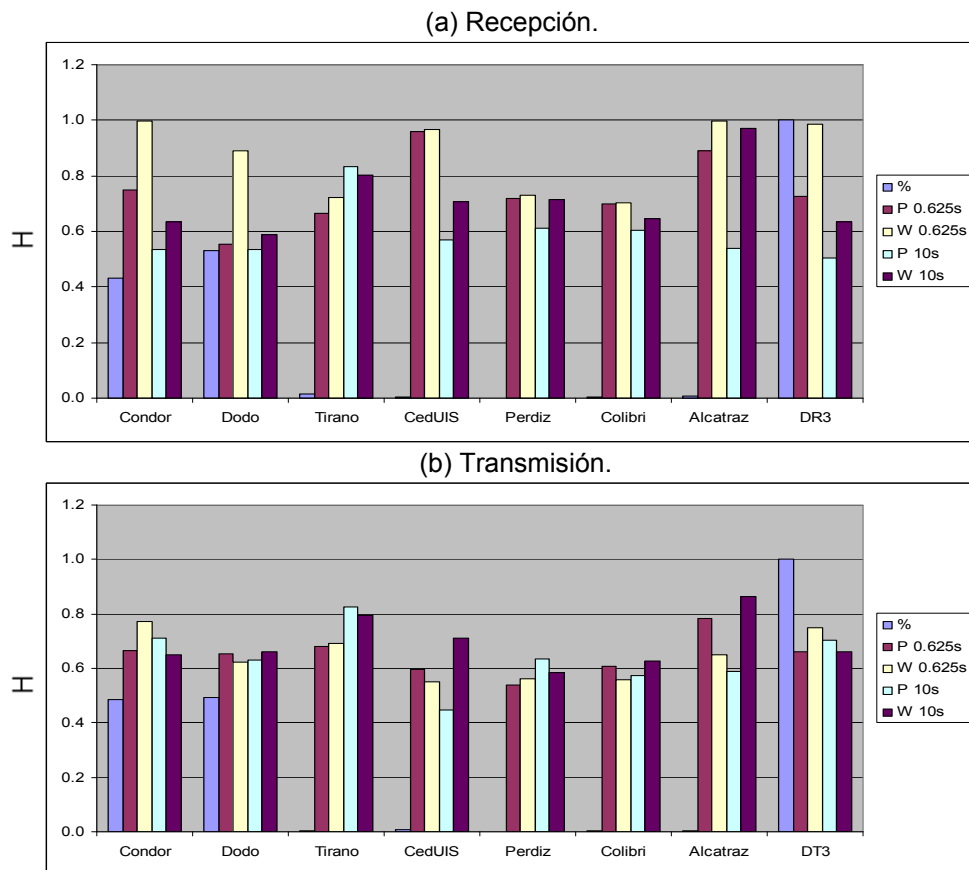


Tabla 11 Resultados prueba 2, para el core switch día 4.

(a) Recepción.								
0.625s	Condor	Dodo	Tirano	CededUIS	Perdiz	Colibri	Alcatraz	Core
kB	389560	590090	27218	2881	2190	5022	53821	1070800
P	0.558	0.528	0.651	0.685	0.774	0.613	0.888	0.546
W	0.750	0.813	0.703	0.811	0.689	0.585	0.895	0.764
10s	Condor	Dodo	Tirano	CededUIS	Perdiz	Colibri	Alcatraz	Core
P	0.584	0.527	0.792	0.602	0.683	0.684	0.722	0.572
W	0.600	0.555	0.753	0.664	0.708	0.628	0.779	0.598
%	0.3638	0.5511	0.0254	0.0027	0.0020	0.0047	0.0503	1.0000
(b) Transmisión.								
0.625s	Condor	Dodo	Tirano	CededUIS	Perdiz	Colibri	Alcatraz	Core
kB	1333900	2074600	21243	13803	15476	9064	88011	3556000
P	0.612	0.593	0.644	0.565	1.226	0.637	0.667	0.618
W	0.719	0.678	0.665	0.768	0.999	0.587	0.603	0.710
10s	Condor	Dodo	Tirano	CededUIS	Perdiz	Colibri	Alcatraz	Core
P	0.605	0.673	0.851	0.576	0.649	0.579	0.684	0.628
W	0.626	0.635	0.731	0.648	0.761	0.631	0.750	0.643
%	0.3751	0.5834	0.0060	0.0039	0.0044	0.0025	0.0248	1.0000

Figura 28 Resultados prueba 2, para el core switch día 4.

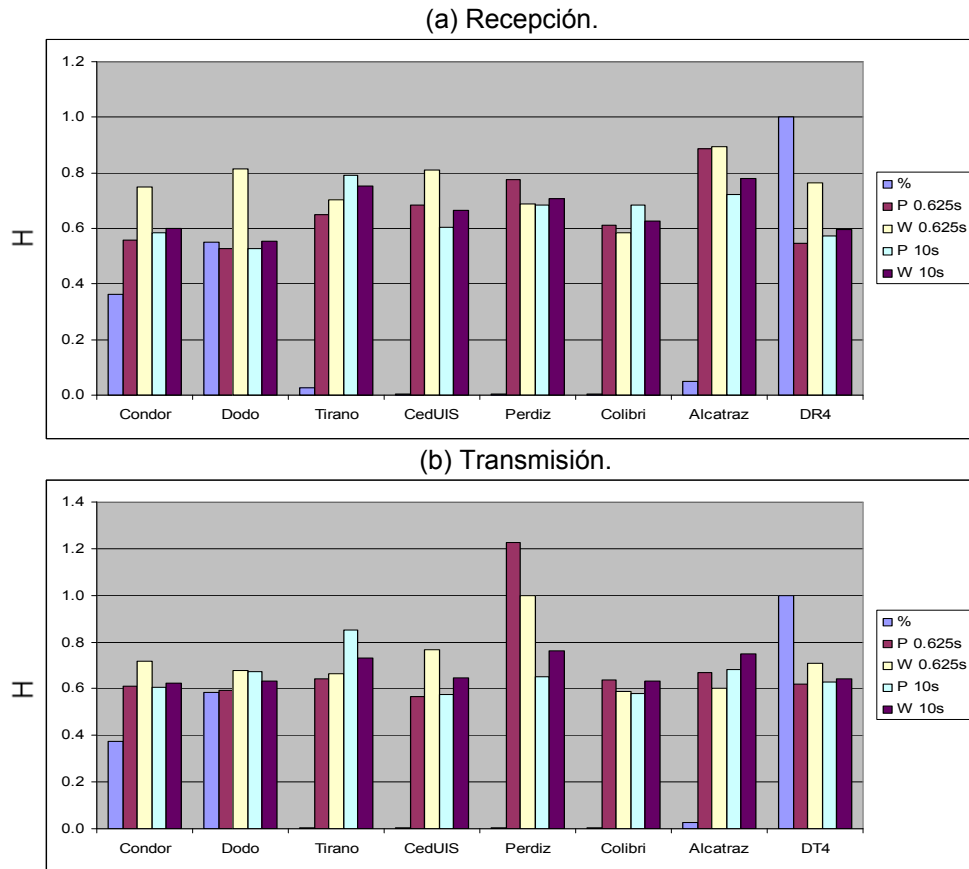


Tabla 12 Resultados prueba 2, para el core switch día 5.

(a) Recepción.								
0.625s	Condor	Dodo	Tirano	CededUIS	Perdiz	Colibri	Alcatraz	Core
kB	487010	465220	52721	2877	1363	1597	29113	1039900
P	0.589	0.541	0.673	0.833	0.753	0.653	0.622	0.583
W	0.983	0.962	0.751	0.999	0.668	0.677	0.941	0.980
10s	Condor	Dodo	Tirano	CededUIS	Perdiz	Colibri	Alcatraz	Core
P	0.692	0.597	1.042	0.606	0.531	0.625	0.894	0.694
W	0.627	0.574	0.919	0.747	0.745	0.693	0.642	0.620
%	0.4683	0.4474	0.0507	0.0028	0.0013	0.0015	0.0280	1.0000
(b) Transmisión.								
0.625s	Condor	Dodo	Tirano	CededUIS	Perdiz	Colibri	Alcatraz	Core
kB	1616000	1491700	32190	11606	9288	3361	63610	3227700
P	0.645	0.682	0.683	0.578	1.223	0.642	0.663	0.654
W	0.877	0.689	0.718	0.577	0.999	0.598	0.617	0.856
10s	Condor	Dodo	Tirano	CededUIS	Perdiz	Colibri	Alcatraz	Core
P	0.768	0.714	1.084	0.706	0.500	0.614	0.863	0.747
W	0.673	0.692	0.918	0.613	0.704	0.654	0.726	0.678
%	0.5007	0.4622	0.0100	0.0036	0.0029	0.0010	0.0197	1.0000

Figura 29 Resultados prueba 2, para el core switch día 5.

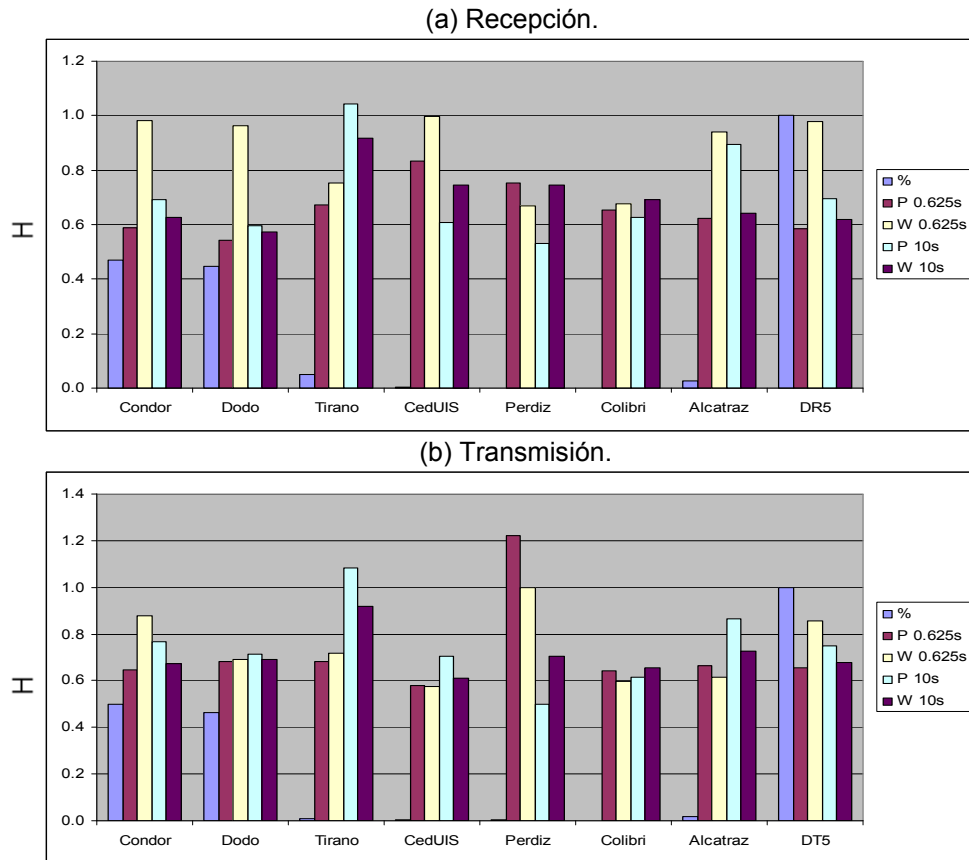
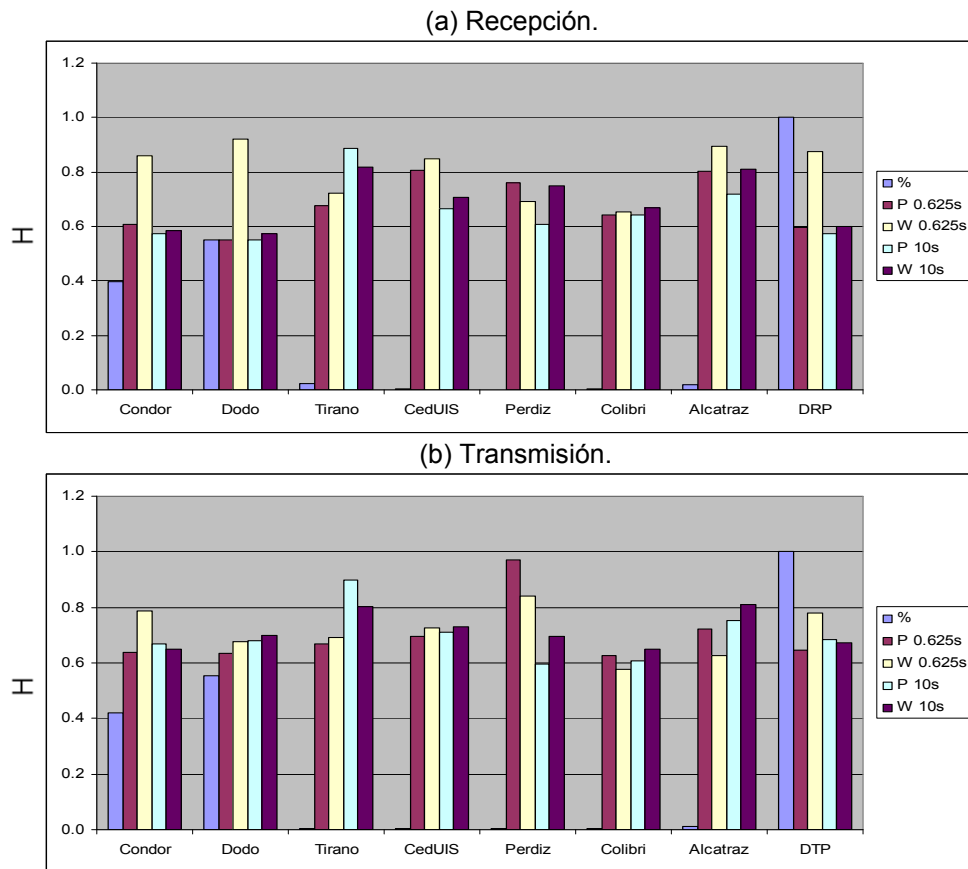


Tabla 13 Resultados prueba 2, para el core switch promediado.

(a) Recepción.								
0.625s	Condor	Dodo	Tirano	CededUIS	Perdiz	Colibri	Alcatraz	Core
kB	463200	638390	27539	3253	1348	3955	22636	1160329
P	0.606	0.550	0.677	0.807	0.762	0.640	0.802	0.598
W	0.859	0.920	0.724	0.847	0.692	0.652	0.893	0.875
10s	Condor	Dodo	Tirano	CededUIS	Perdiz	Colibri	Alcatraz	Core
P	0.574	0.552	0.888	0.667	0.606	0.643	0.717	0.572
W	0.584	0.572	0.817	0.709	0.747	0.670	0.809	0.599
%	0.3992	0.5502	0.0237	0.0028	0.0012	0.0034	0.0195	1.0000
(b) Transmisión.								
0.625s	Condor	Dodo	Tirano	CededUIS	Perdiz	Colibri	Alcatraz	Core
kB	1650000	2166900	19172	20807	8429	8336	45181	3918794
P	0.638	0.636	0.669	0.694	0.973	0.625	0.723	0.645
W	0.788	0.675	0.692	0.727	0.842	0.578	0.628	0.780
10s	Condor	Dodo	Tirano	CededUIS	Perdiz	Colibri	Alcatraz	Core
P	0.669	0.679	0.900	0.709	0.597	0.608	0.752	0.683
W	0.648	0.698	0.804	0.732	0.695	0.651	0.810	0.673
%	0.4210	0.5530	0.0049	0.0053	0.0022	0.0021	0.0115	1.0000

Figura 30 Resultados prueba 2, para el core switch promediado.



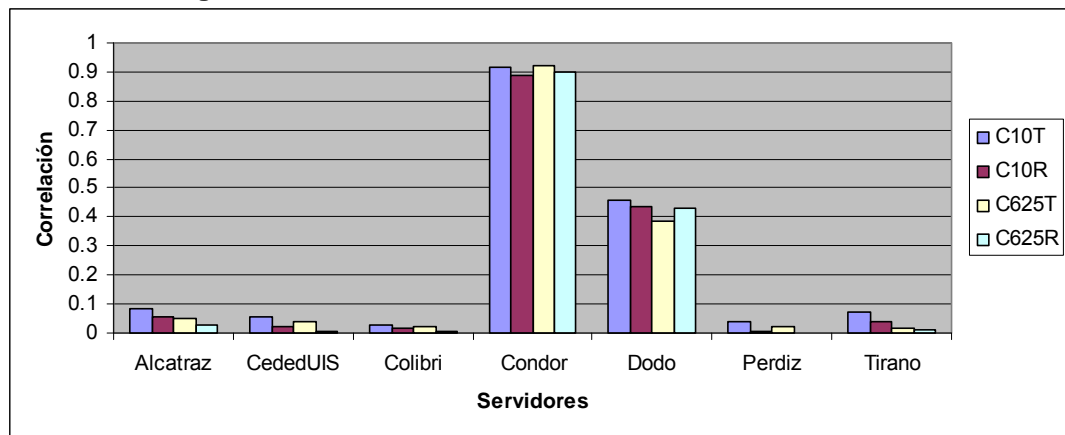
De la primera prueba se observa que tanto el periodograma como el estimador de Whittle disminuyen a medida que el grano aumenta de tamaño, esto es de esperarse debido a que al aumentar el tiempo de la prueba la autosimilitud entre los datos se degrada tal como se observa en la figura 12.

De la segunda prueba se obtiene que un grano más fino presenta mejores resultados en autosimilitud aunque no es muy notoria en el periodograma. Sin embargo, se destaca de la prueba que la autosimilitud del servidor Cónдор con el core switch es muy cercana o similar.

4.4 Observaciones de la red

Para demostrar la conclusión de la sección anterior, se utiliza un análisis del coeficiente de correlación entre los servidores y el core switch; se hace individualmente por días y después se promedia. La figura 31 muestra el grado de correlación que existe entre el servidor Cónдор y el *core switch*, en la figura: C para correlación, 10 y 625 para el tamaño del grano, T y R transmisión y recepción respectivamente.

Figura 31 Correlación de los servidores contra el core switch.



A pesar que el servidor Cónдор no es el que presenta mayor flujo de datos, dicta la manera como se comportan los datagramas en el core switch para los demás servidores. Esto es interesante ya que al conocer que servidores tienen mayor influencia sobre el tráfico, el análisis de nodos importantes se simplifica.

4.5 Aleatoriedad en la red

Se desarrolla además, un análisis probabilístico de la red a partir del cual se puedan generar datos en forma aleatoria y observar si la red realmente debe modelarse en forma autosimilar o probabilística.

NOTA: Teniendo en cuenta que los servidores Cóndor y Dodo aportan más del 95% de las muestras de tráfico tomadas en el *core switch*, solamente se obtendrán las características del comportamiento probabilístico para los servidores Cóndor, Dodo, y para el *core switch*; este tipo de análisis requiere de un alto procesamiento matemático, y teniendo en cuenta que las demás fuentes no hacen un aporte significativo sobre el *core*, se pueden considerar despreciables.

Para determinar la PDF^{10} de los servidores y el core switch, se deben realizar pruebas de bondad de ajuste, las más conocidas son la K-S (sección 4.5.1) y la Chi 2 (sección 4.5.2).

Se determina con las pruebas de bondad de ajuste que el servidor Cóndor tiene una PDF con forma log-logística, mientras que el servidor Dodo y el Core Switch tienen una PDF con forma Burr. Esto podría significar que probabilísticamente el core switch y el servidor Cóndor no guardan una relación tan estrecha como se plantea en la sección 4.4, pero es necesario aclarar que la PDF log-logística es un caso particular de Burr.

La tabla 14 presenta los resultados probabilísticos para los servidores Cóndor, Dodo y el core switch. En las tablas, k , α y β , son los parámetros de las PDF para cada día (la distribución log-logística no tiene parámetro k). Al final de la tabla se presentan los promedios (P) para cada parámetro. Adicionalmente, en gris se presentan los datos que no se ajustaron a dicho análisis de bondad. Como es necesario estandarizar a un solo modelo probabilístico estos datos no se utilizan para calcular el promedio.

La ecuación que define la PDF Burr (ecuación 4.3) y la PDF log-logística (ecuación 4.4), se presentan a continuación.

¹⁰ Por sus siglas en ingles Probability Density Function.

Tabla 14 Resultados probabilísticos: Condor, Dodo, Core switch.

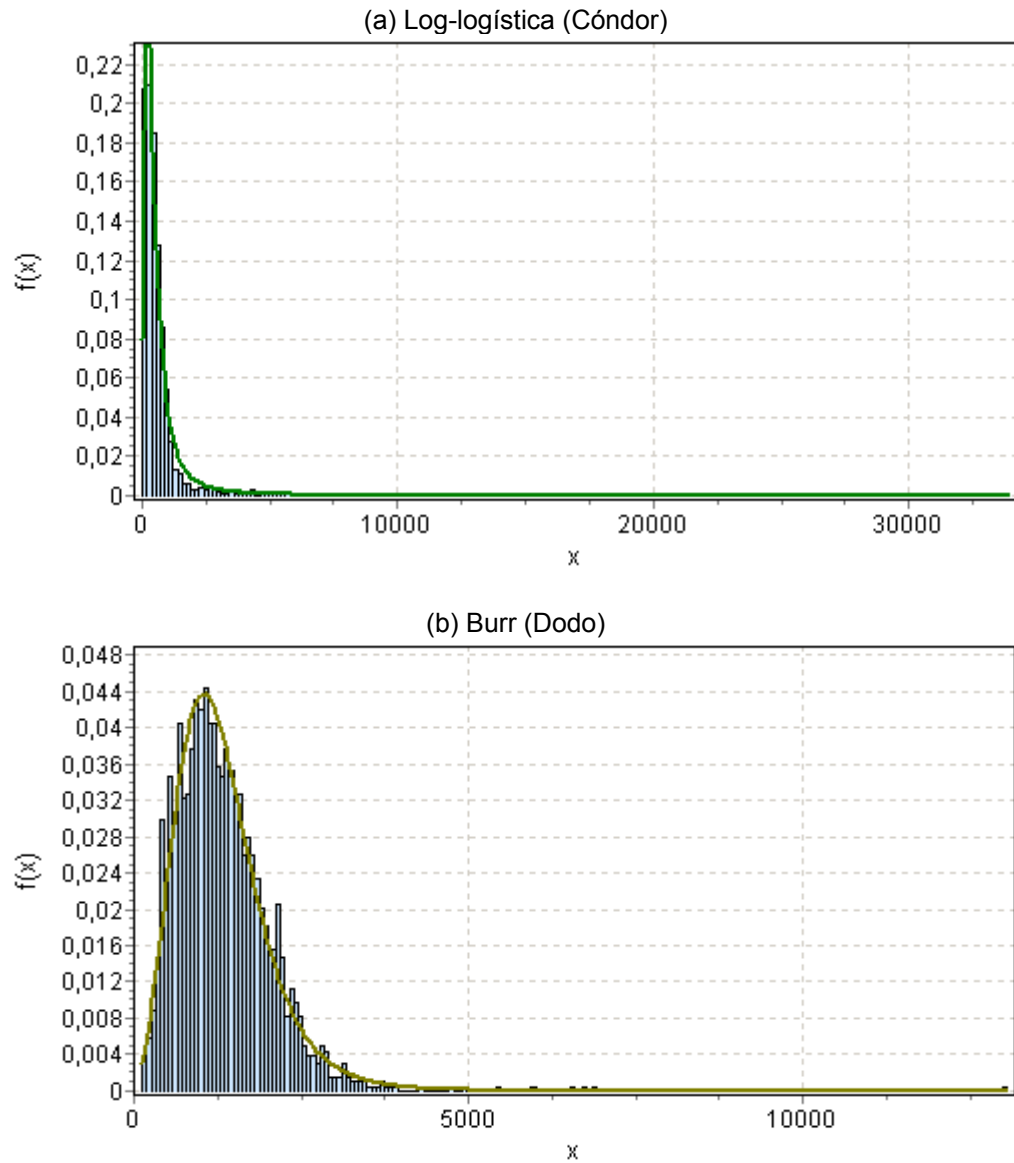
	Condor(T)	Condor(R)	Dodo(T)	Dodo(R)	Core(T)	Core(R)
k			2.6036	1.6813	0.8229	0.4487
alpha	1.8082	1.532	2.5993	2.9284	3.4593	4.4305
beta	432.95	87.196	1926.5	363.06	1642.3	298.45
k			4.4462	2.1599	1.8251	1.0044
alpha	1.4831	2.111	2.4641	2.6563	2.4263	3.2114
beta	374.09	65.012	2224.7	507.94	2283.6	442.63
k			1.6051	1.6589	0.6916	0.6275
alpha	1.3784	1.7454	2.6286	2.5581	2.9156	3.509
beta	411.31	80.761	1154	339.44	1248.9	304.8
k			0.0026	1.7748	0.6742	0.7215
alpha	1.7853	1.9165	497.35	2.4743	3.6675	3.4365
beta	340.79	79.039	175.75	320.38	1157	321.12
k			2.8133	1.9977	1.0433	0.5798
alpha	1.8002	2.0744	2.0991	2.158	2.8118	4.1131
beta	445.44	109.07	1167.8	274.4	1281.6	280.61
(P) k			2.8671	1.8545	1.0114	0.6764
(P) alpha	1.651	1.8759	2.4478	2.555	3.0561	3.7401
(P) beta	400.92	84.216	1618.3	361.04	1522.7	329.52

$$f(x) = \frac{ak \left(\frac{x}{\beta}\right)^{a-1}}{\beta \left(1 + \left(\frac{x}{\beta}\right)^a\right)^{k+1}} \quad (4.3)$$

$$f(x) = \frac{a}{\beta} \left(\frac{x}{\beta}\right)^{a-1} \left(1 + \left(\frac{x}{\beta}\right)^a\right)^{-2} \quad (4.4)$$

En la figura 32 se presentan las *PDF* log-logística (los datos pertenecen al servidor Cónдор) y Burr (los datos pertenecen al servidor Dodo). En las figuras en ambos casos los datos se agrupan a la izquierda, por lo cual, las ráfagas de tráfico de mayor amplitud ocurren raras veces. Se advierte además, que la distribución log-logística es muy similar a la distribución Burr, se diferencian principalmente en el valor de la media y la varianza.

Figura 32 Función de densidad de probabilidad de los servidores Cónдор y Dodo.



4.5.1 Prueba Kolmogorov-Smirnov (K-S) [18]

La prueba K-S se utiliza únicamente en casos de variables continuas (aquella que toma un número infinito de valores entre n y $n+1$), y compara una *PDF* empírica contra la *PDF* correspondiente a los datos a evaluar, utilizando la ecuación 4.1.

La prueba cuenta con el hecho que la función de densidad acumulativa está distribuida de una forma normalmente asintótica.

$$D = \max_{1 \leq i \leq n} |\hat{F}(x_i) - F(x_i)| \quad (4.1)$$

Donde,

x_i es el i -ésimo valor observado en la muestra.

$\hat{F}(x_i)$ es la *PDF* empírica.

$F(x_i)$ es la *PDF* con los datos muestreados.

El resultado final de la prueba representa el parecido entre las distribuciones de los datos. A menor D , mayor es su parecido.

4.5.2 Prueba Chi-cuadrado (Chi 2) [18]

La prueba Chi 2 se puede utilizar para casos de variable continua y variable discreta. Ésta prueba es utilizada para determinar si una muestra viene de una población con distribución hipotética.

El análisis consiste en agrupar los datos con base en el tamaño de la prueba. El cálculo para el tamaño de los grupos es muy variado.

La ecuación 4.2 define la prueba Chi 2.

$$x^2 = \sum_{i=1}^k \frac{(O_i - E_i)^2}{E_i} \quad (4.2)$$

Donde,

O_i es la frecuencia observada para el grupo i

E_i es la frecuencia calculada para el grupo i

k es el número de grupos

El resultado final de la prueba representa el parecido entre las distribuciones de los datos. A menor x^2 , mayor es su parecido.

5. SIMULACIÓN DE LA RED

La simulación permite utilizar las relaciones matemáticas y probabilísticas que describen un proceso real, con el fin de evaluarlo en un ambiente controlado, y así, estudiar los efectos producidos por alteraciones en el modelo del sistema.

Por medio de la simulación se pueden anticipar resultados no previstos, detección de variables significativas de las cuales depende directamente el sistema, observar escenarios que tal si, experimentar con posibles mejoras, entre otras ventajas que incluyen un costo reducido y una fracción del tiempo que requiere hacer pruebas en sistemas reales.

Existen muchos simuladores de tráfico de redes, los más utilizados se exponen en el capítulo 2 del presente documento.

5.1 Escenario de simulación y ajustes

La figura 33 muestra el escenario de simulación en formato nam de ns2, la tabla 15 presenta un resumen de las funciones de cada nodo. El código en TCL utilizado se encuentra en el anexo C.

Figura 33 Escenario de simulación en ns2.

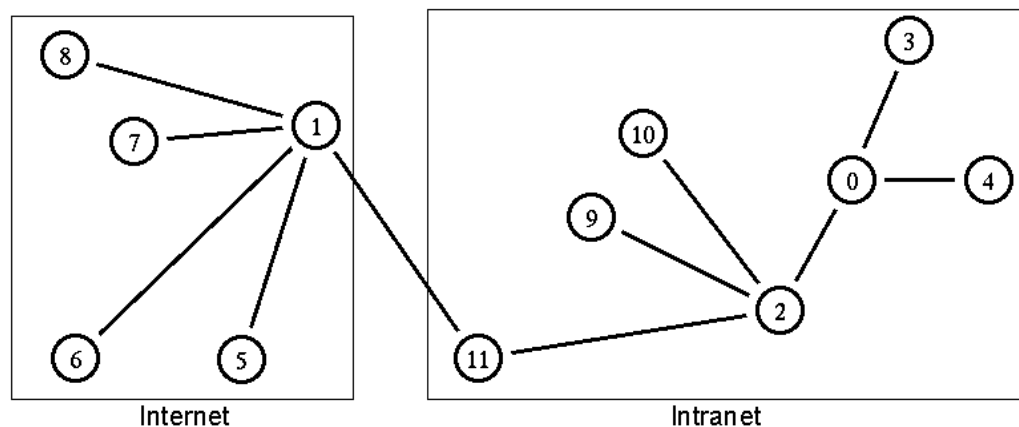


Tabla 15 Descripción de los nodos del escenario de simulación.

Nodo	Función
0	Switch DSI, conecta los servidores a la Intranet.
1	Switch clientes Internet.
2	Core switch, conecta los servidores y los clientes de la Intranet.
3	Servidor Condor (SMTP).
4	Servidor Dodo (HTTP).
5-10	Clientes.
11	Router, conecta la Intranet a la Internet.

Las características de ancho de banda y retardos utilizados en la simulación son aproximados a los presentados en el capítulo 4. Se expone en detalle las características del simulador a continuación:

Todas las conexiones de la Internet utilizan un ancho de banda aleatorio entre 1Mb/s y 2Mb/s, y tienen retardos aleatorios entre 10ms y 30ms.

Todas las conexiones de la Intranet UIS utilizan un ancho de banda de un Gigabit/s, y utilizan un retardo de 1 ms. Al nodo 2 se conectan los clientes UIS, al nodo 0 se conectan los servidores UIS.

El nodo 11 encargado de conectar la Intranet UIS con la Internet es la única excepción con respecto a las conexiones anteriormente descritas. Este nodo se conecta al nodo 2 con ancho de banda de un Gigabit/s, pero con retardo de 3ms; a su vez se conecta con el nodo 1 con un ancho de banda de 10Mb/s (el real es de 30Mb/s pero se debe tener en cuenta que esta conexión se comparte con otras aplicaciones que utilizan la conexión) y un retardo de 10ms.

El nodo 3 se utiliza para simular el tráfico correspondiente a SMTP mediante la generación, en los nodos cliente, una combinación de tráfico de protocolo Telnet y FTP, donde Telnet se asocia al envío del texto del correo electrónico y FTP a los adjuntos. Las peticiones se hacen en forma aleatoria. Esta implementación aproximada se utiliza en el trabajo en vista que el simulador utilizado no ofrece un módulo para SMTP explícito.

El nodo 4 que corresponde a un servidor HTTP, contiene un grupo de páginas con características similares a las del portal web UIS real. El nodo está programado para recibir peticiones por parte de los nodos cliente. Las peticiones se generan en forma aleatoria.

Variables servidor SMTP:

Este servidor utiliza 5 variables aleatorias; 2 (\$tamanoftpi, \$tiempoftpi) se utilizan para controlar los adjuntos que se envían al servidor de correos y 2 (\$tamanoftpo, \$tiempoftpo) para solicitar el adjunto, cada par maneja el tamaño del archivo (distribución tipo Pareto) y la frecuencia (distribución tipo exponencial) a la cual se adjunta o se solicita un archivo; la última variable aleatoria (\$intervalo) controla la frecuencia (distribución tipo exponencial) con que se realizan las conexiones Telnet. Estas distribuciones típicas para la frecuencia y el tamaño de los archivos se encuentran descritas en la página 46 de [9].

La distribución de Pareto utiliza 2 parámetros para su configuración el promedio (avg_) y la forma (shape_) también conocida como β_i cuyo valor se utiliza para calcular H en la ecuación 3.5, sin embargo, éste β_i toma valores entre 1 y 2 por lo cual es necesario reemplazar en la ecuación 3.5 $\beta = \beta_i - 1$. Los valores de β_i se ajustan entre 1.7 y 1.8 lo que corresponde a un valor de H entre 0.6 y 0.65.

El parámetro del promedio debe ser configurado conjuntamente con la frecuencia de envío/solicitud de adjuntos y la frecuencia de conexiones Telnet. Para el servidor cónдор la transmisión promedio proveniente de los clientes es de aproximadamente 20kB/s y hacia los clientes es de 80kB/s. Se utiliza la ecuación 5.1 para determinar el promedio de la transmisión, donde $1/\alpha$ es la frecuencia (α es el parámetro avg_ de la distribución exponencial en segundos).

$$\bar{P} = \frac{1}{\alpha} (\text{tamaño archivo en bytes}) * (\# \text{ clientes}) \quad (5.1)$$

Para la transmisión de datos desde los clientes al nodo servidor 3, se generan adjuntos con un α igual a 30s y tamaño de archivo de 80KB para 6 clientes, con lo cual se obtiene un promedio de transmisión de adjuntos de 16KB/s. A éste resultado se le debe sumar el tráfico generado por las conexiones Telnet para lo cual se utiliza un α igual a 8s y un tamaño de archivo que oscila entre los 3KB (por defecto) para cada uno de los 6 clientes, logrando un promedio de transmisión de 2.25KB/s. La transmisión de los clientes al nodo servidor 3 es de 18.25KB/s, se debe adicionar además un aproximado del 10% que corresponde al tráfico generado por los acuses, obteniendo una transmisión promedio total de 20.075KB/s.

La transmisión de datos desde el nodo servidor 3 hacia los clientes utiliza un α igual a 30s y un tamaño de archivo de 350KB para 6 clientes, lo cual genera un tráfico de 70KB/s, que sumado al tráfico generado por los acuses y las solicitudes Telnet que corresponde aproximadamente al 10%, conduce a un promedio total de 77KB/s.

Para comprobar lo anteriormente señalado, se obtienen por separado las transmisiones de los adjuntos, las transmisiones Telnet y las transmisiones de los acuses. Se encuentra que sí se ajusta.

Variables servidor HTTP:

Este servidor utiliza 10 variables aleatorias, 6 son utilizadas para generar páginas de tamaño aleatorio y las 4 restantes para obtener las características de la página principal. La configuración de la simulación se basa en el tamaño de las páginas UIS reales. En [21] se explica detalladamente el comportamiento de cada una de las variables del tráfico HTTP.

Las páginas aleatorias utilizan las siguientes variables:

\$intersesion tiempo entre sesiones de clientes, en segundos (exponencial).
\$numpag consultas realizadas por sesión (uniforme).
\$interpag frecuencia con que el cliente ingresa a una nueva página de la sesión, en segundos (exponencial).
\$stampag cantidad de objetos que contiene la nueva página (constante).
\$interobj tiempo entre envío de objetos, en segundos (exponencial).
\$tamobj número de paquetes por objeto, paquetes de 1kB (Pareto).

Para las páginas aleatorias se configuran las variables con los siguientes valores:

\$intersesion	7s
\$numpag	2 a 6
\$interpag	10s
\$stampag	8 objetos
\$interobj	0.01s
\$tamobj	5 con $\beta_i = 1.5$

Con estos datos se obtienen páginas web con tamaño promedio de 40kB, el parámetro β_i se ajusta para tener una autosimilitud aproximada de 0.7.

Para la página principal de la UIS se configuran las siguientes variables aleatorias:

\$intersesion	5s
\$numpag	0
\$interpag	0s
\$stampag	80 objetos
\$interobj	0.01s
\$tamobj	5 con $\beta_i = 1.5$

Con lo cual se obtiene una página con tamaño promedio de 400kB.

A diferencia del servidor SMTP en este caso se genera una transmisión a la vez (aunque la solicitud la hace un cliente), y no depende directamente de la cantidad de clientes, se tiene entonces que el tráfico generado por las páginas aleatorias es 40kB/7s más un promedio de 4 sesiones que genera cada 10 segundos (4x40kB/10s), más el tráfico que genera la página principal que es de 400kB/5s, por lo cual el tráfico generado es de 101.7kB/s, más el tráfico que aportan los acuses aproximado a un 10%, se obtiene que el tráfico total es de 112kB/s aproximadamente en sentido servidor-cliente. La tasa de transferencia promedio empírica en el sentido servidor-cliente es de 110kB/s.

El tráfico proveniente de los clientes al servidor es generado automáticamente y no requiere ser calculado. La tasa de transferencia promedio empírica en el sentido cliente-servidor es de 25kB/s.

5.2 Autosimilitud de la simulación

Para la simulación se utilizan las mismas pruebas realizadas al tráfico real con el propósito de comparar los resultados posteriormente.

A continuación se muestran (prueba 1: tabla 16, figura 34; prueba 2: tabla 17, figura 35) los resultados del periodograma y el estimador de Whittle para determinar la autosimilitud del tráfico agregado en el nodo 0 y cada uno de los nodos servidor, de acuerdo con las pruebas mencionadas en la sección 4.2.

Se separan las tablas y las figuras en (a) y (b), donde (a) son los resultados para los datos recibidos y (b) los resultados para los datos transmitidos, P y W hacen referencia a periodograma y estimador de Whittle respectivamente.

La tabla 16 presenta los resultados para la prueba 1. D hace referencia a los resultados para un grano de 10 segundos, H# para el grano de 1 segundo, y M# para el grano de 0.1 segundos.

La tabla 17 presenta los resultados para la prueba 2. 0,625s y 10s hacen referencia a los tamaños de grano, kB es el tráfico transmitido durante la prueba.

La primera prueba presenta características similares a las obtenidas con los datos empíricos, donde el periodograma y el estimador de Whittle disminuyen con el aumento del tamaño del grano.

En el caso de la segunda prueba, se logran mejores resultados en autosimilitud con la disminución del tamaño del grano. Además la simulación muestra que los

valores de autosimilitud del nodo 4 son más próximos a los del nodo 0, contrario a lo sucedido en las pruebas empíricas.

Tabla 16 Resultados prueba 1, para el tráfico agregado del nodo 0.

(a) Recepción.											
H#	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	D
P	0.660	0.540	0.669	0.678	0.580	0.533	0.654	0.543	0.629	0.511	0.528
W	0.715	0.800	0.627	0.654	0.654	0.641	0.689	0.657	0.722	0.727	0.638
M#	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	
P	0.867	0.734	0.905	0.829	0.716	0.580	0.827	0.719	0.875	0.829	
W	0.851	0.717	0.776	0.887	0.805	0.995	0.751	0.814	0.879	0.808	
(b) Transmisión.											
H#	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	D
P	0.523	0.657	0.643	0.680	0.632	0.655	0.992	0.569	0.670	0.618	0.650
W	0.873	0.730	0.703	0.917	0.682	0.690	0.942	0.692	0.669	0.861	0.678
M#	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	
P	1.180	0.811	0.735	1.240	0.706	1.103	1.370	0.742	0.572	0.675	
W	0.998	0.947	0.801	0.996	0.784	0.988	0.998	0.881	0.859	0.868	

Figura 34 Resultados prueba 1, para el tráfico agregado del nodo 0.

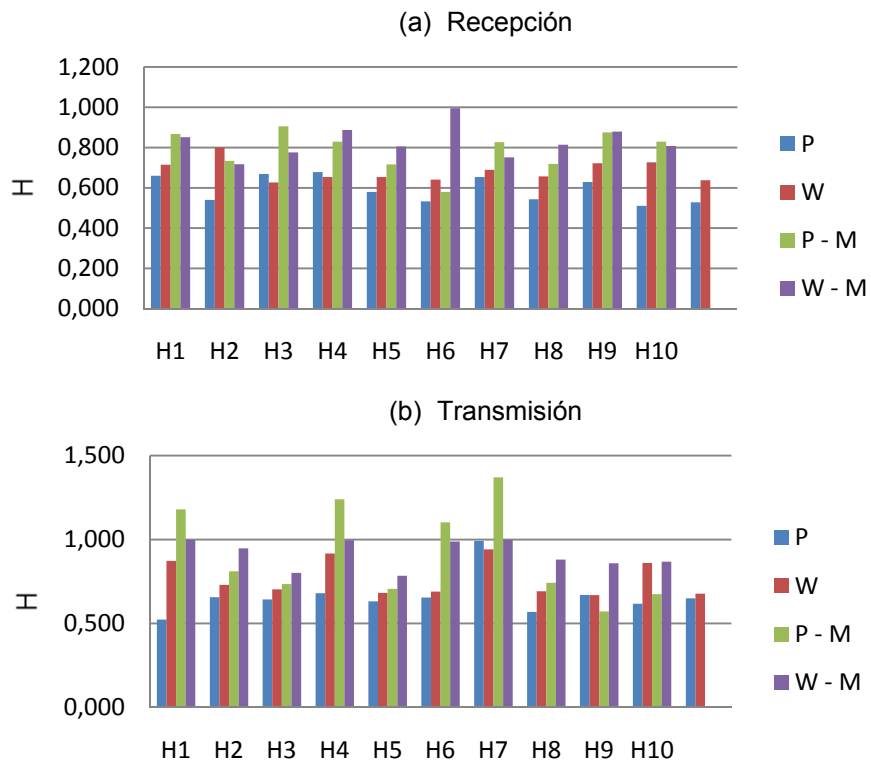
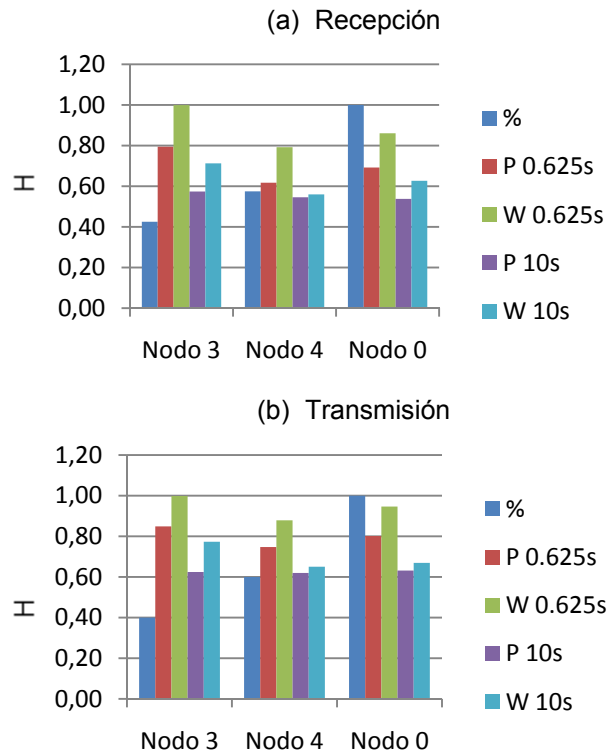


Tabla 17 Resultados prueba 2, para el tráfico agregado del nodo 0.

(a) Recepción							
Nodo ↓	Prueba →	1	2	3	4	5	Promedio
3 (10s)	KB	395630	410220	380500	442510	390140	403800
	P	0.574	0.598	0.542	0.576	0.579	0.574
	W	0.718	0.721	0.697	0.711	0.715	0.712
	%	0.42	0.43	0.41	0.45	0.42	0.43
3 (0.625s)	P	0.801	0.806	0.790	0.789	0.785	0.794
	W	0.999	0.999	0.999	0.999	0.999	0.999
4 (10s)	KB	545710	545950	549990	537320	548110	545416
	P	0.541	0.534	0.554	0.571	0.530	0.546
	W	0.557	0.558	0.562	0.562	0.560	0.560
	%	0.58	0.57	0.59	0.55	0.58	0.57
4 (0.625s)	P	0.673	0.621	0.580	0.611	0.601	0.617
	W	0.835	0.784	0.778	0.783	0.780	0.792
0 (10s)	KB	941340	956170	930490	979830	938250	949216
	P	0.545	0.528	0.533	0.535	0.549	0.538
	W	0.637	0.638	0.618	0.623	0.618	0.627
0 (0.625)	P	0.676	0.723	0.683	0.688	0.690	0.692
	W	0.839	0.864	0.866	0.868	0.866	0.861
(b) Transmisión							
Nodo ↓	Prueba →	1	2	3	4	5	Promedio
3 (10s)	KB	1338400	1489800	1392000	1517400	1618400	1471200
	P	0.638	0.681	0.596	0.607	0.601	0.625
	W	0.794	0.801	0.755	0.762	0.755	0.773
	%	0.38	0.40	0.38	0.42	0.42	0.40
3 (0.625s)	P	0.845	0.859	0.835	0.853	0.853	0.849
	W	0.999	0.998	0.999	0.999	0.999	0.999
4 (10s)	KB	2202800	2242900	2255700	2061050	2272000	2206890
	P	0.698	0.615	0.627	0.560	0.599	0.620
	W	0.674	0.661	0.628	0.675	0.614	0.650
	%	0.62	0.60	0.62	0.58	0.58	0.60
4 (0.625s)	P	0.586	0.864	0.661	0.845	0.780	0.747
	W	0.775	0.931	0.835	0.974	0.879	0.879
0 (10s)	KB	3541200	3732700	3647700	3578450	3890400	3678090
	P	0.716	0.650	0.601	0.589	0.601	0.631
	W	0.691	0.678	0.663	0.674	0.640	0.669
0 (0.625)	P	0.812	0.867	0.696	0.841	0.790	0.801
	W	0.999	0.952	0.873	0.996	0.912	0.946

Figura 35 Resultados prueba 2, para el tráfico agregado del nodo 0.



La tabla 18 muestra los promedios en autosimilitud de los nodos 3, 4 y 0 contra los promedios en autosimilitud empíricos de los servidores Cóndor, Dodo y el Core. Note que para un grano de 10 segundos los resultados son muy similares entre la simulación y los datos empíricos. El grano pequeño presenta una mayor diferencia entre los dos casos.

Tabla 18 Promedios autosimilitud simulados contra empíricos prueba 2.

(a) Recepción						
	Nodo 3	Cóndor	Nodo 4	Dodo	Nodo 0	Core
P - 10s	0.574	0.574	0.546	0.552	0.538	0.572
W - 10s	0.712	0.584	0.560	0.572	0.627	0.599
P - 0.625s	0.794	0.606	0.617	0.550	0.692	0.598
W - 0.625s	0.999	0.859	0.792	0.920	0.860	0.875
(b) Transmisión						
	Nodo 3	Cóndor	Nodo 4	Dodo	Nodo 0	Core
P - 10s	0.625	0.669	0.620	0.679	0.631	0.683
W - 10s	0.773	0.648	0.650	0.698	0.669	0.673
P - 0.625s	0.849	0.638	0.747	0.636	0.801	0.645
W - 0.625s	0.999	0.788	0.879	0.675	0.946	0.780

5.3 Aleatoriedad de la simulación

Se realizan las pruebas de bondad de ajuste (sección 4.5) a los resultados de la simulación. Se encuentra que el nodo 0 tiene una *PDF* tipo Burr, el nodo 3 presenta una *PDF* de tipo log-logística y para el nodo 4 se obtiene una *PDF* de tipo Burr.

La tabla 19 presenta los resultados probabilísticos para los nodos 0, 3 y 4. En la tabla “error %” corresponde al valor del error cuadrático medio de los datos simulados respecto a la distribución empírica (sección 4.5). El error se calcula mediante la prueba K-S.

La figura 36 (a) muestra la *PDF* promedio para el servidor Cóndor sobre el histograma obtenido en una simulación del nodo 3. Se observa que el primer valor del histograma no cumple con la *PDF* teórica (esto ocurre en algunos casos teóricos del servidor Cóndor), por lo cual es necesario segmentar la *PDF* en trozos como se muestra en la ecuación 5.2. Con esto se obtienen resultados muy próximos entre el tráfico empírico y el simulado. El error cuadrático medio de esta prueba es calculado filtrando los datos en donde $x < 20kB$.

$$f(x) = \begin{cases} 10\% = 0 & ; x < 20kB \\ 90\% = teórica & ; x > 20kB \end{cases} \quad (5.2)$$

La figura 36 (b) muestra la *PDF* promedio para el servidor Dodo sobre el histograma obtenido de una simulación del nodo 4. Se presenta en este caso una distribución con parámetros muy distantes a los empíricos, sin embargo, un análisis del error cuadrático medio (menor al 10%) demuestra que los resultados son muy próximos a la distribución empírica.

El resultado final muestra que el nodo 0 presenta un error cuadrático medio de casi el 5% con respecto a la distribución empírica, teniendo en cuenta que el simulador tiene 15 variables aleatorias, se demuestra que a través de una simulación se pueden obtener resultados probabilísticos muy próximos a los reales.

Tabla 19 Resultados probabilísticos nodos: 0, 3 y 4.

	Nodo 3 (T)	Nodo 3 (R)	Nodo 4 (T)	Nodo 4 (R)	Nodo 0 (T)	Nodo 0 (R)
k			6.02	428.00	2.88	1.91
alpha	1.82	2.12	1.87	1.81	2.32	3.01
beta	237.00	97.00	3338.00	8561.00	2429.00	538.00
error %	5.40	7.10	9.37	8.60	8.70	6.20
k			2.86	22.67	2.06	1.80
alpha	1.74	2.01	1.80	1.83	2.53	3.01
beta	311.00	115.00	1823.00	1622.00	2124.00	525.50
error %	3.30	5.85	8.51	8.06	4.10	6.80
k			9.83	408.00	3.60	2.07
alpha	1.29	2.29	1.55	1.80	2.41	2.91
beta	405.00	129.00	5070.00	8637.00	3040.00	562.44
error %	6.10	8.63	8.70	8.05	4.50	6.02
k			2.98	22.50	1.89	1.97
alpha	1.28	2.12	1.63	1.81	2.65	2.94
beta	490.00	89.00	1774.00	1605.00	2100.00	543.00
error %	5.40	7.33	9.27	9.17	3.39	5.28
k			3.72	35.00	2.32	2.02
alpha	1.32	2.33	1.72	1.81	2.66	2.95
beta	475.00	109.00	2300.00	2115.00	2501.00	557.00
error %	4.10	6.78	7.90	7.95	8.39	6.38
(P) k			5.08	183.23	2.55	1.95
(P) alpha	1.49	2.17	1.71	1.81	2.51	2.96
(P) beta	383.60	107.80	2861.00	4508.00	2438.80	545.19
(P) error %	4.86	7.14	8.75	8.37	5.82	6.14

5.4 Observaciones de la simulación

La figura 37 presenta el coeficiente de correlación de los nodos servidor (3 y 4) con respecto al nodo de tráfico agregado (0), con el fin de observar si ocurre el mismo fenómeno en que un servidor con menor volumen de tráfico puede tener mayor influencia sobre el punto de tráfico agregado. En la figura: C para correlación, 10 y 625 para el tamaño del grano, T y R transmisión y recepción respectivamente.

En 5.2 se mencionó que los resultados en autosimilitud del nodo 4 son más cercanos a los obtenidos en el nodo 0, como se demuestra con el estudio de la correlación en la figura 37. Esto es contrario a la prueba empírica, por lo cual el simulador falló en mantener ésta propiedad del segmento de red estudiado.

Figura 36 Histogramas de los nodos 3 y 4.

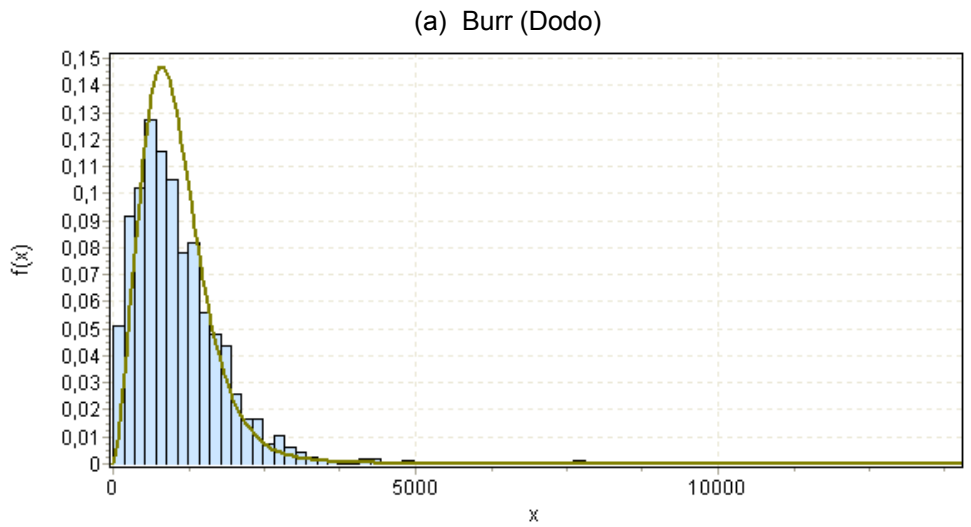
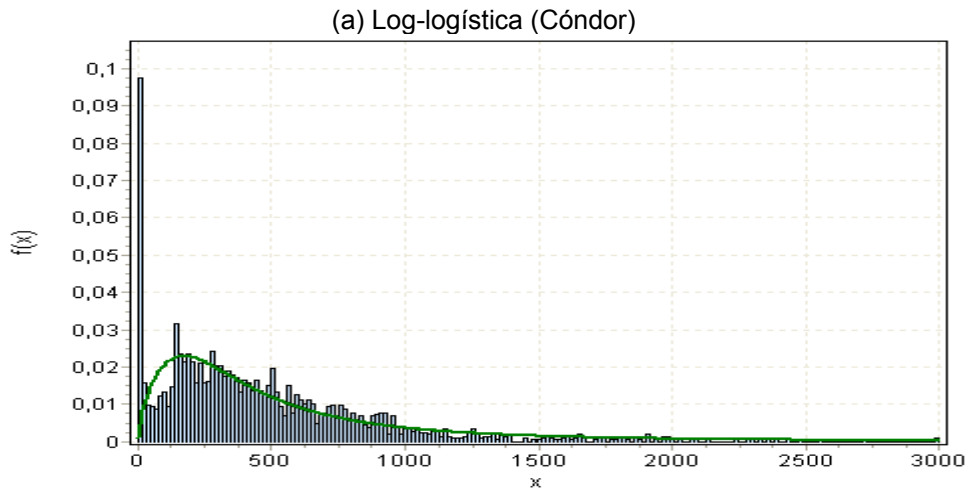
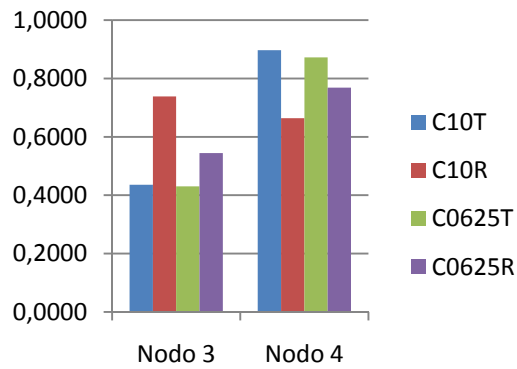


Figura 37 Correlación de los nodos servidor contra el nodo de tráfico agregado.



CONCLUSIONES

El parámetro de Hurst puede ser obtenido mediante una variedad de estimadores. En este trabajo en particular el periodograma y el estimador de Whittle permitieron calcular resultados que se ubican dentro de los límites teóricos (0,5 1]. Otros estimadores suelen sobrepasar estos límites con frecuencia.

Sobre el cálculo y uso del parámetro de Hurst se encontraron varios aspectos valiosos. En la prueba 1, a medida que la escala de tiempo aumenta el valor del parámetro disminuye. Esta situación es limitante para el estudio de la dependencia a largo plazo y no permite caracterizar el tráfico con ese sólo parámetro; en la prueba 2, el tamaño del grano presenta notorias diferencias en los resultados obtenidos con el estimador de Whittle, sin embargo, los resultados del periodograma son más próximos entre sí, teniendo en cuenta esta propiedad del periodograma no se hace necesario utilizar grano pequeño para determinar el valor del parámetro de Hurst. Esta propiedad es muy importante ya que permite hacer el estudio de un nodo con un menor costo computacional.

Debido a que muchos sistemas pueden presentar resultados similares en autosimilitud, es necesario hacer un estudio para obtener las características probabilísticas del tráfico, el cual permite conocer su distribución, su media y desviación, datos muy importantes para ajustar las variables aleatorias del simulador.

La correcta simulación del modelo depende de la herramienta de simulación seleccionada, ns2 tiene excelentes opciones que facilitan el diseño de simulaciones a pesar de utilizar línea de código y no una interfaz gráfica. Con el NS-2 se logró generar tráfico cuyas características de autosimilitud resultaron muy próximas a aquellas de los registros de tráfico empíricos. Adicionalmente, el valor del parámetro de Hurst calculado a partir de las trazas de tráfico simuladas, tiene relación directa con las modificaciones realizadas al tamaño de los paquetes en cualquiera de los dos tipos de tráfico simulados.

Las simulaciones mostraron para la prueba 1 que a medida que el tamaño del grano aumenta la autosimilitud disminuye, y para la prueba 2 que al disminuir el tamaño del grano el estimador de Whittle genera valores más altos de autosimilitud y que el periodograma es más consistente en sus resultados a pesar del tamaño del grano. Las dos pruebas conservan las mismas propiedades tanto en el tráfico empírico como en el tráfico simulado.

Mediante el cálculo del error cuadrático medio se estimó la diferencia entre las trazas de tráfico generadas por simulación y las distribuciones que mejor

ajustaban los datos experimentales. La prueba mostró un error menor al 5% para el servidor SMTP (nodo 3), cercano al 8% para el servidor HTTP (nodo 4) y cercano al 5% para el nodo 0, teniendo en cuenta que el sistema utiliza 15 variables aleatorias se considera que los resultados obtenidos con la simulación son excelentes.

BIBLIOGRAFÍA

- [1] A. Popescu. *Traffic self-similarity*. in Proc. IEEE Int. Conf. Telecommunications, Bucharest, Romania, 1993.
- [2] M. Taqqu and Teverovsky. *On estimating the intensity of long-range dependence in finite and infinite variance time series*.
- [3] W. Leland and D. Wilson. *High time-resolution measurement and analysis of lan traffic: Implications for lan interconnection*. Bellcore Labs, 1990.
- [4] D. McDysan. *QoS and Traffic Management in IP and ATM Networks*. McGraw-Hill, 2000.
- [6] W. Willinger, W. Leland, M. Taqqu and D. Wilson. *On the self-similar nature of ethernet traffic*. Proceedings of ACM SIGCOMM 93, 1993.
- [7] F. Pallikari, E. Boller. *A rescaled range analysis of random events*. Journal of scientific exploration, Vol. 13, No. 1, pp 25 – 40, 1999.
- [8] W. Castellanos, M. Chacón. *Fundamentos teóricos y herramientas software para el modelado y la simulación de redes de comunicaciones*. Unidades tecnológicas de Santander. 2005.
- [9] E. Altman, T. Jiménez. *NS Simulator for Beginners*. Universidad de los Andes, Mérida, Venezuela, and ESSI Sophia-Antipolis, France. 2003.
- [10] A. Montoya, *Antenas fractales: un paso en la evolución de las telecomunicaciones*, Universidad de Antioquia, Colombia.
- [11] Página oficial de Adventnet 6.
<http://www.adventnet.com/products/simulator/help/index.html>
- [12] Página oficial de Adventnet 6.
http://www.adventnet.com/products/simulator/demo6/adventnet_netsim.htm
- [13] Página oficial de Network Simulator
<http://www.isi.edu/nsnam/>

- [14] Página oficial de OpNet, práctica 1: Evaluating Internet Connection Choices for a small Home PC Network.
http://www.opnet.com/university_program/teaching_with_opnet/textbooks_and_materials/itg_panko.html
- [15] A. Varga, Using the OMNeT++ Discrete Event Simulation System in Education.
<http://www.ewh.ieee.org/soc/es/Nov1999/18/Begin.htm>
- [16] Página oficial de Cisco
http://www.wvup.edu/dweaver/Cisco/PacketTracer4/packet_tracer_module/packet_tracer_module/default.htm#
- [17] Página oficial de Shunra
<http://www.shunra.com/vedesktop.aspx>
- [18] Página oficial de EasyFit – Distribution Fitting Software
<http://www.mathwave.com/>
- [19] Thomas Karagiannis, Michalis Faloutsos, Mart Molle. A User-Friendly Self-Similarity Analysis Tool. Special Section on Tools and Technologies for Networking Research and Education, ACM SIGCOMM Computer Communication Review, 2003.
- [20] Página oficial Wireshark
<http://www.wireshark.org/>
- [21] M. Molina, P. Castelli, G. Foddis, *Web traffic modeling exploiting TCP connections temporal clustering through HTML-REDUCE*, IEEE, 2000.
- [22] FRAS, Matjaz, MOHORKO, Joze. Estimating the parameters of measured self similar traffic for modeling in OPNET. V: CUCEJ, Zarko (ur.), PLANINSIC, Peter (ur.), GLEICH, Dusan (ur.). 2007 IWSSIP & EC-SIPMCS: CD proceedings of 2007 14th International Workshop on Systems, Signals and Image Processing (IWSSIP) and 6th EURASIP Conference Focused on Speech & Image Processing, Multimedia Communications and Services (EC-SIPMCS), 27.-30 June 2007, Maribor, Slovenija.
- [23] Lee I.C.W. and Fapojuwo A.O. "Stochastic Processes for Computer Network Traffic Modeling," Computer Communications Journal, vol. 29, Issue 1, December 2005, pp. 1 - 23.

ANEXO A: CONFIGURACIÓN DEL WIRESHARK

El parámetro de Hurst del *core switch* se encuentra a través del análisis de los datagramas compartidos. Para adquirir dichos datagramas es necesario utilizar un *sniffer* de tráfico como Wireshark, el cual es gratuito.

A.1. Adquisición de datos con Wireshark

En Wireshark es importante escoger que parámetros de los paquetes son necesarios para un análisis de autosimilaridad, para esto vaya a *Edit >> Preferences >> Columns* y borre con el botón *Delete* cada uno de los contenidos presentes. Con el botón *New* se creará una con el título ***Time*** y formato ***Time (format as specified)***, la segunda será ***Length*** y formato ***Packet Length (bytes)***, y por último ***Source*** con formato ***Source address***. Asegúrese que queden en el orden: *Time*, *Length* y *Source*, como se muestra en la figura 38. Una vez finalizado haga clic en *Apply* y luego en *Ok*, y reinicie wireshark para que los cambios tengan efecto.

Seguido debe asegurar que Wireshark capture datos en modo promiscuo, para esto vaya a *Capture >> Options* y seleccione *capture packets in promiscuous mode*. Para evitar las direcciones MAC utilice en *Capture Filter*: ***not arp.opcode <= 0x002***, asegúrese que en *Display options* y en *Name Resolution* ninguna opción esté seleccionada. Además, en muchas ocasiones es posible llenar la memoria disponible para Wireshark, para evitar esto de clic en la opción *Use multiple files*, *Next file every 400 MBs*, *Stop capture after 6 hours*. De clic en *Start* para empezar a capturar datos, observe la figura 39. Si no captura, cambie en *Interface* a la tarjeta de red que está utilizando.

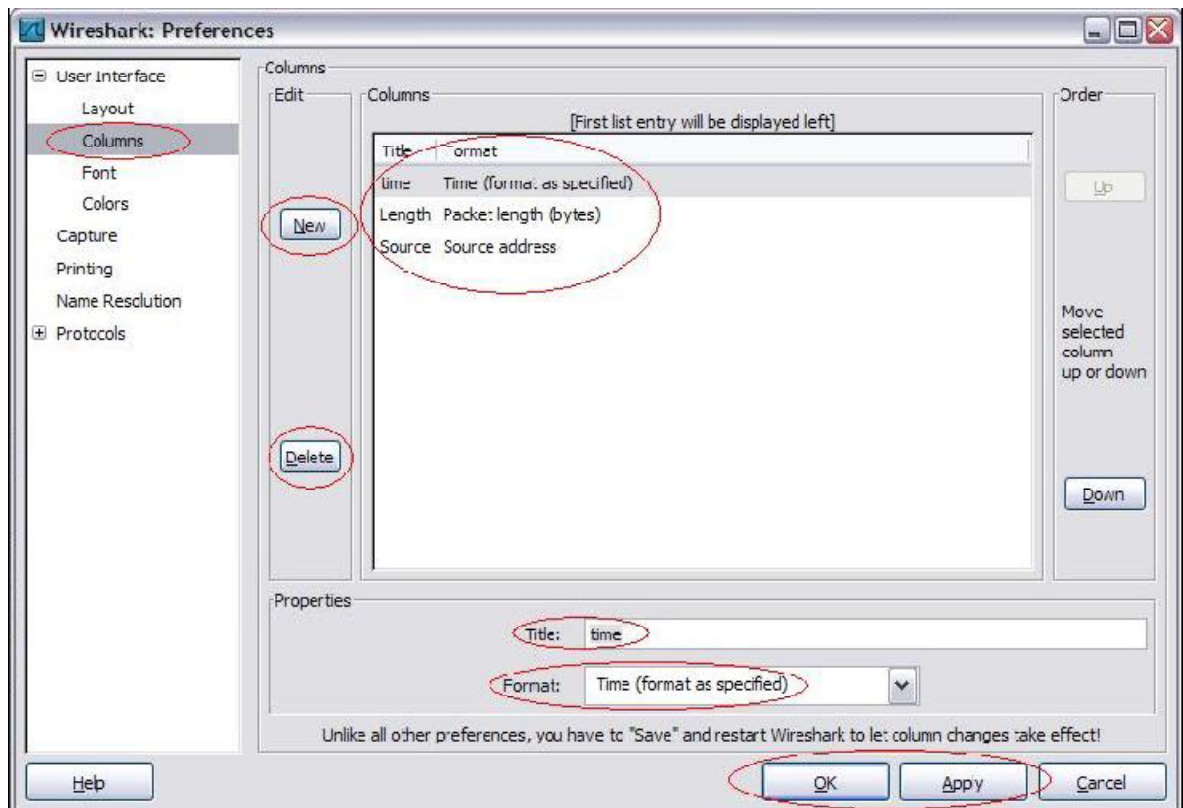
Nota 1: En el caso de hacer un análisis de tráfico agregado el filtro a utilizar es: ***not arp.opcode <=0x002 and (192.168.19.2 or 192.168.19.15 or 192.168.19.70 or ...)***, utilice las direcciones IP correspondientes a los servidores que está capturando.

Una vez finalizada la captura vaya a *File >> Export >> File*, en *Packet Format* asegúrese que *Packet details* no esté seleccionado, escoja un nombre.txt y guarde, como se muestra en la figura 40.

No olvide guardar la dirección IP del servidor, es importante.

Nota 2: Cuando se generan múltiples archivos, se deben abrir en Wireshark y salvar uno a uno como se describió anteriormente.

Figura 38 Edición de las preferencias de Wireshark.



Ahora abra el archivo nombre.txt con un software de procesamiento de texto (ejemplo: Microsoft Word), elimine la primera línea: *time length source*. En buscar vaya a reemplazar, digite la dirección IP del servidor que desea analizar y reemplácela por un número superior a 255, esto para poder utilizar los *scripts* de Matlab del anexo B.

A.2 Análisis con la herramienta Selfis

Una vez terminado el proceso de conversión con Matlab los archivos obtenidos serán utilizados en Selfis [19]. De clic en la carpeta y cargue el archivo, seguido de clic en play para obtener los resultados del parámetro de Hurst. Como se muestra en la figura 41.

Figura 39 Opciones de captura en Wireshark.

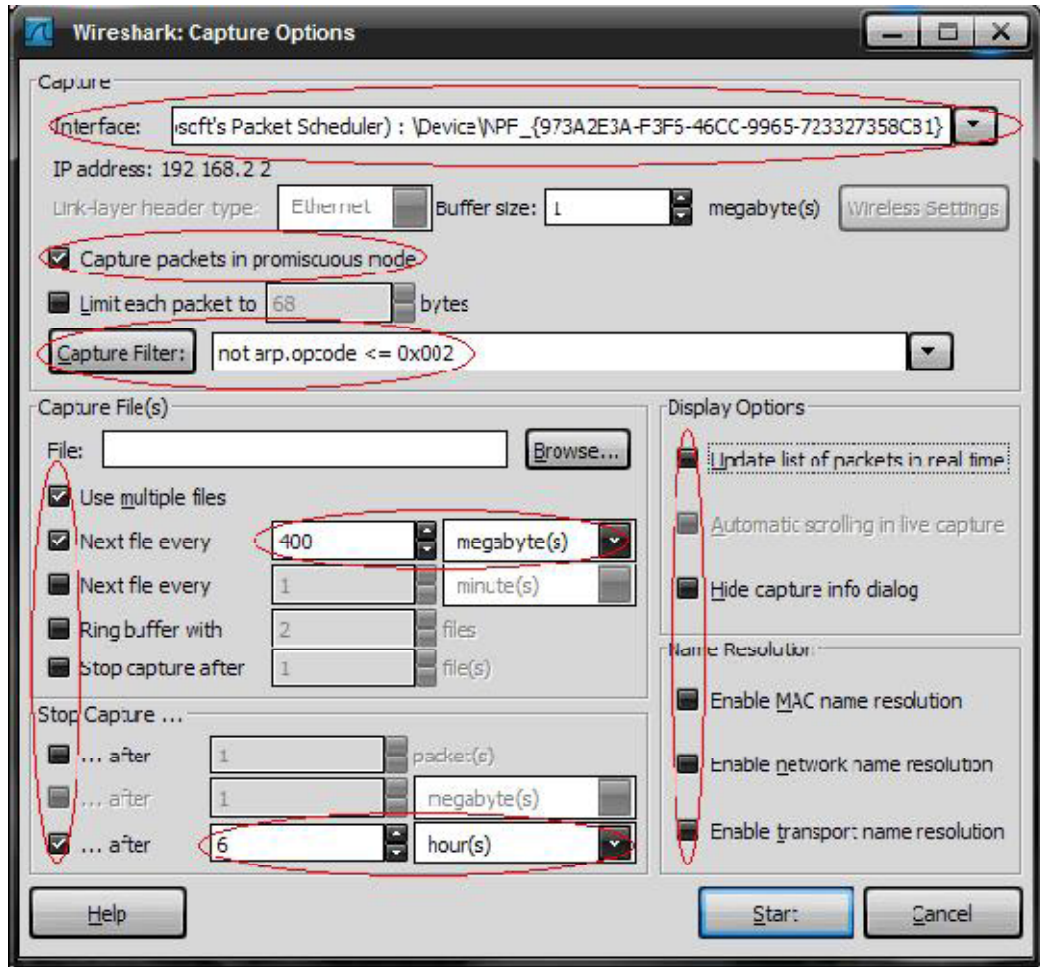


Figura 40 Pantalla para exportar a *.txt.

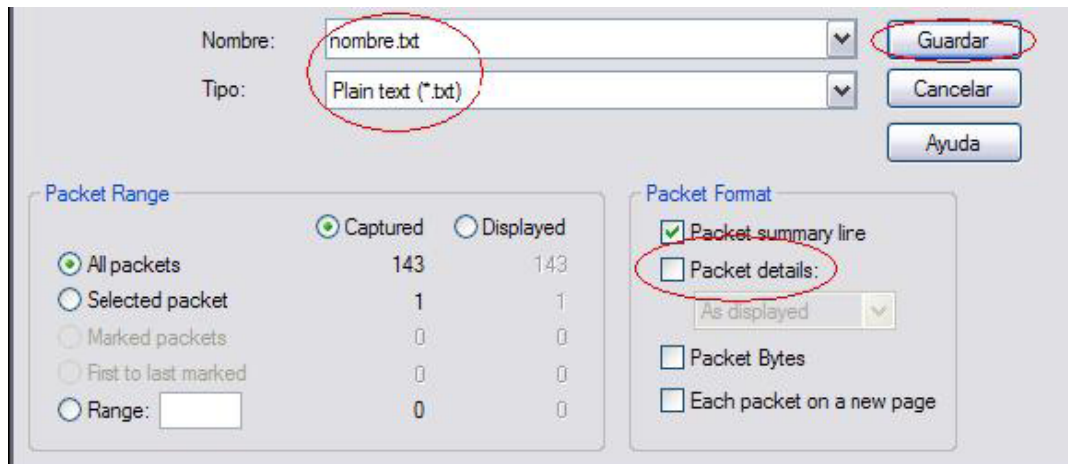
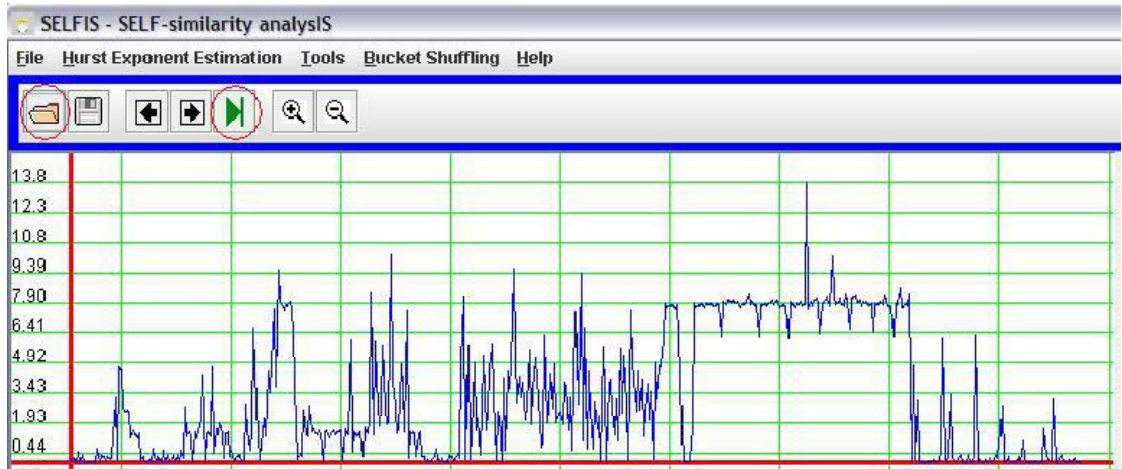


Figura 41 Pantalla principal de Selfis.



ANEXO B: CÓDIGOS PARA LAS PRUEBAS

B.1 Código para la prueba 1 (MATLAB)

```
%% Este script transforma datagramas tomados con wireshark, a varios
%% vectores de tráfico en kBytes/(unidad de tiempo) contra tiempo, con las
%% características de esta investigación.

%% Con este script se obtienen los vectores para la prueba 1:
%% - vector de entrada/salida de datos para granos de 0.1/1/10
%% segundos para el core switch.
%% - volumen de datos transmitidos y recibidos para cada vector.

%% NOTA: Leer anexo A sobre la obtención de los archivos en el formato
%% necesario para utilizar este script. Lea además, las notas del script
%% para configurar entradas, y reconocer los vectores que almacenan volumen
%% de tráfico (el volumen se entrega en kBytes).

%% -----
%% Entradas

%% Al ejecutar este script se le solicitará primero el número de archivos
%% que componen el datagrama, y posteriormente se le pedirán los nombres de
%% los archivos.

%% El script tiene un ejemplo en comentario sencillo (%), para utilizarlo
%% debe comentar la línea 29 y 37, y quitar el comentario sencillo (%) a
%% las líneas que lo tengan (29 - 71). Debe reemplazar en la línea 30 el
%% número de bucles con el número de archivos que componen el datagrama,
%% entre las líneas 38 - 71 para el comando (R = load('swff1##.txt'))
%% utilizar los nombres de los archivos en .txt que componen el datagrama.

bucles = input ('Archivos en Wireshark que componen la prueba: ');
%bucles = 16

filaAB = 1; %contador filas vectores
A = 0; %vector temporal entradas
B = 0; %vector temporal salidas

for cont1 = 1:bucles %bucle para solicitud de
    R = load(input ('Abrir Archivo: ')); %archivos.
% switch cont1
% case 1
% R = load('swff101.txt');
% case 2
% R = load('swff102.txt');
% case 3
% R = load('swff103.txt');
```

```

% case 4
% R = load('swff104.txt');
% case 5
% R = load('swff105.txt');
% case 6
% R = load('swff106.txt');
% case 7
% R = load('swff107.txt');
% case 8
% R = load('swff108.txt');
% case 9
% R = load('swff109.txt');
% case 10
% R = load('swff110.txt');
% case 11
% R = load('swff111.txt');
% case 12
% R = load('swff112.txt');
% case 13
% R = load('swff113.txt');
% case 14
% R = load('swff114.txt');
% case 15
% R = load('swff115.txt');
% case 16
% R = load('swff116.txt');
% end

```

```

%% -----
%% Este segmento utiliza 1 vector temporal en el cual se procesa la
%% información de los 7 servidores, 990 a 996 corresponde a la dirección IP
%% reemplazada debido a que matlab no procesa ese tipo de datos.

```

```

%% Vectores con grano de 0.1 segundos

```

```

filas = size(R,1);
a = 0;
b = 0;
time = 0.1;
t = time;

for cont2 = 1:filas
    if R(cont2,1) < t
        if R(cont2,3) < 900
            a = a + R(cont2,2);
        else
            b = b + R(cont2,2)
        end
    else
        A(filaAB) = a/1000;
        B(filaAB) = b/1000;
        cont2 = cont2 - 1;
        filaAB = filaAB + 1;
        t = t + time;
    end
end

```

```

        a = 0;
        b = 0;
    end
end
end

%% Vectores temporales con unidad de tiempo 1 segundo y 10 segundos
%%(A01s y B01s para 1s, A10s y B10s para 10s).

A01s = 0; B01s = 0; A10s = 0; B10s = 0;
t01 = 1; t10 = 1;

for cont1 = 1:2048,
    A01s(cont1) = sum(A(t01:t01+9));
    B01s(cont1) = sum(B(t01:t01+9));
    A10s(cont1) = sum(A(t10:t10+99));
    B10s(cont1) = sum(B(t10:t10+99));
    t01 = t01 + 10;
    t10 = t10 + 100;
end

%% -----
%% Segmento para salvar los vectores de entrada y salida del core switch,
%% con grano de 10 segundos y 1 segundo.

input ('Archivos con grano de 10 segundos')

done = input ('Guardar entrada con grano de 10s como: ');
datos = A10s(1:2048);
dlmwrite (done, datos)
done = input ('Guardar salida con grano de 10s como: ');
datos = B10s(1:2048);
dlmwrite (done, datos)

input ('10 segmentos con grano de 1 segundo')

t = 1;
VservA = 0;
VservB = 0;

for i = 1:10
    done = input ('Guardar entrada con grano de 1s como: ');
    datos = A01s(t:t+2047);
    dlmwrite (done, datos)
    VservA(i) = sum(datos);
    done = input ('Guardar salida con grano de 1s como: ');
    datos = B01s(t:t+2047);
    dlmwrite (done, datos)
    VservB(i) = sum(datos);
    t = t + 2048;
end

%% -----
%% Volumen de tráfico para los vectores con grano de 1 segundo y 10

```

```

%% segundos.

VservA(11) = sum(A10s);
VservB(11) = sum(B10s);

%% -----
%% Segmento para salvar los vectores de entrada y salida del core switch,
%% con grano de 0.1 segundos.

204800
t = 10001;
VAm = 0;
VBm = 0;

for i = 1:10
    done = input ('Guardar entrada con grano de 0.1s como: ');
    datos = A(t:t+2047);
    dlmwrite (done, datos)
    VAm(i) = sum(datos);
    done = input ('Guardar salida con grano de 0.1s como: ');
    datos = B(t:t+2047);
    dlmwrite (done, datos)
    VBm(i) = sum(datos);
    t = t + 20480;
end

%% -----
%% El volumen de tráfico para los vectores con grano de 0.1s se encuentra
%% almacenado en la variable VAm y VBm.
%% -----

```

B.2 Código para la prueba 2 (MATLAB)

```

%% Este script transforma datagramas tomados con wireshark, a varios
%% vectores de tráfico en kBytes/(unidad de tiempo) contra tiempo, con las
%% características de esta investigación.

%% Con este script se obtienen los vectores para la prueba 2:
%% - vector de entrada/salida de datos para unidad de tiempo de 0.625/10
%% segundos para cada servidor y el core switch.
%% - volumen de datos transmitidos y recibidos por cada servidor y el core
%% switch.

%% NOTA: Leer anexo A sobre la obtención de los archivos en el formato
%% necesario para utilizar este script. Lea además, las notas del script
%% para configurar entradas, y reconocer los vectores que almacenan volumen
%% de tráfico (el volumen se entrega en kBytes).

%% -----
%% Entradas

```

```

%% Al ejecutar este script se le solicitará primero el número de archivos
%% que componen el datagrama, y posteriormente se le pedirán los nombres de
%% los archivos.

```

```

bucles = input('Archivos en Wireshark que componen la prueba: ');

```

```

filaAB = 1; %contador filas vectores
A = [0, 0, 0, 0, 0, 0, 0]; %vector temporal entradas
B = [0, 0, 0, 0, 0, 0, 0]; %vector temporal salidas

```

```

for cont1 = 1:bucles %bucle para solicitud de
    R = load(input('Abrir Archivo: ')); %archivos.

```

```

%% -----
%% Este segmento utiliza una matriz temporal en la cual se procesa la
%% información de los 7 servidores, 990 a 996 corresponde a la dirección IP
%% reemplazada debido a que matlab no procesa ese tipo de datos.

```

```

%% Vectores con grano de 0.625 segundos

```

```

filas = size(R,1);
a = [0, 0, 0, 0, 0, 0, 0];
b = [0, 0, 0, 0, 0, 0, 0];
time = 0.625;
t = time;

```

```

for cont2 = 1:filas
    if R(cont2,1) < t
        if R(cont2,3) < 900
            switch R(cont2, 4)
                case 990
                    a(1) = a(1) + R(cont2,2);
                case 991
                    a(2) = a(2) + R(cont2,2);
                case 992
                    a(3) = a(3) + R(cont2,2);
                case 993
                    a(4) = a(4) + R(cont2,2);
                case 994
                    a(5) = a(5) + R(cont2,2);
                case 995
                    a(6) = a(6) + R(cont2,2);
                case 996
                    a(7) = a(7) + R(cont2,2);
            end
        else
            switch R(cont2,3)
                case 990
                    b(1) = b(1) + R(cont2,2);
                case 991
                    b(2) = b(2) + R(cont2,2);
            end
        end
    end
end

```

```

        case 992
            b(3) = b(3) + R(cont2,2);
        case 993
            b(4) = b(4) + R(cont2,2);
        case 994
            b(5) = b(5) + R(cont2,2);
        case 995
            b(6) = b(6) + R(cont2,2);
        case 996
            b(7) = b(7) + R(cont2,2);
        end
    end

else
    A(filaAB,1:7) = a/1000;
    B(filaAB,1:7) = b/1000;
    cont2 = cont2 - 1;
    filaAB = filaAB + 1;
    t = t + time;
    a = [0, 0, 0, 0, 0, 0, 0];
    b = [0, 0, 0, 0, 0, 0, 0];
end
end
end

%% Vectores temporales con grano de 10 segundos (As, Bs)

As = [0, 0, 0, 0, 0, 0, 0];
Bs = [0, 0, 0, 0, 0, 0, 0];
t = 1;

for cont1 = 1:2048,
    As(cont1,1:7) = sum(A(t:t+15,1:7));
    Bs(cont1,1:7) = sum(B(t:t+15,1:7));
    t = t + 16;
end

%% -----
%% Segmento para salvar los vectores de entrada y salida de los 7
%% servidores, con grano de 0.625 segundos y 10 segundos.

input ('Servidores individuales')

for i = 1:7
    done = input ('Guardar entrada de 0.625 seg como: ');
    datos = A(1:32768, i);
    dlmwrite (done, datos)
    done = input ('Guardar salida de 0.625 seg como: ');
    datos = B(1:32768, i);
    dlmwrite (done, datos)
end

```

```

for i = 1:7
    done = input ('Guardar entrada de 10 seg como: ');
    datos = As(1:2048, i);
    dlmwrite (done, datos)
    done = input ('Guardar salida de 10 seg como: ');
    datos = Bs(1:2048, i);
    dlmwrite (done, datos)
end

%% Segmento para salvar los vectores de entrada y salida del core switch,
%% con grano de 0.625 segundos y 10 segundos.

input ('Core central')

ACS = sum(A, 2);
BCS = sum(B, 2);
ACSs = sum(As, 2);
BCSs = sum(Bs, 2);

done = input ('Guardar entrada de 0.625 seg como: ');
datos = ACS(1:32768);
dlmwrite (done, datos)
done = input ('Guardar salida de 0.625 seg como: ');
datos = BCS(1:32768);
dlmwrite (done, datos)

done = input ('Guardar entrada de 10 seg como: ');
datos = ACSs(1:2048);
dlmwrite (done, datos)
done = input ('Guardar salida de 10 seg como: ');
datos = BCSs(1:2048);
dlmwrite (done, datos)

%% -----
%% Volumen de tráfico en los servidores
VservA = sum(As, 1);
VservB = sum(Bs, 1);

%% Volumen de tráfico en el core switch
VcsA = sum(VservA);
VcsB = sum(VservB);
%% -----

```

ANEXO C: CÓDIGOS DEL SIMULADOR

C.1 Simulador Intranet UIS

```
set ns [new Simulator]
puts "Para detener la simulación: Ctrl+c"

# Archivos traza
$ns trace-all [open uislan.out w]
$ns namtrace-all [open uislan.nam w]

***** CONFIGURACIÓN
#----- Variables topografía
set tiempo 100

set clientes 40
set clientesuis 2
set servidores 2
set tnodos [expr 4 + [expr $clientes + $servidores]]

#Secciones HTTP (no modificar)
set sesiones 20000

#----- Variables aleatorias
#Telnet (intervalo entre sesiones)
set intervalo [new RandomVariable/Exponential]
$intervalo set avg_ 8

#FTP clientes - servidor (tiempo entre sesiones)
set tiempoftpi [new RandomVariable/Exponential]
$tiempoftpi set avg_ 30
#FTP clientes - servidor (tamano sesiones)
set tamanoftpi [new RandomVariable/Pareto]
$tamanoftpi set avg_ 80000
$tamanoftpi set shape_ 1.7

#FTP servidor - clientes (tiempo entre sesiones)
set tiempoftpo [new RandomVariable/Exponential]
$tiempoftpo set avg_ 30
#FTP servidor - clientes (tamano sesiones)
set tamanoftpo [new RandomVariable/Pareto]
$tamanoftpo set avg_ 350000
$tamanoftpo set shape_ 1.8

#HTTP Intervalo entre sesiones
set interseccion [new RandomVariable/Exponential]      ;# Páginas secundarias
$interseccion0 set avg_ 7
set interseccion2 [new RandomVariable/Exponential]    ;# Página principal
$interseccion1 set avg_ 5
#----- Nodos
for {set i 0} {$i < $tnodos} {incr i} {
    set n($i) [$ns node]
}
```

```

#----- Conexiones
#Servidores y routers
$ns duplex-link $n(0) $n(2) 1000Mb 1ms DropTail
$ns duplex-link $n(2) $n([expr $tnodos - 1]) 1000Mb 3ms DropTail
$ns duplex-link $n([expr $tnodos - 1]) $n(1) 10Mb 10ms DropTail

for {set i 0} {$i < $servidores} {incr i} {
    $ns duplex-link $n(0) $n([expr $i + 3]) 1000Mb 1ms DropTail
}
#Clientes Internet
for {set i 0} {$i < $clientes} {incr i} {
    if {$i < [expr $clientes - $clientesuis]} {
        set BW [uniform 2 20]
        set retardo [uniform 10 100]
        $ns duplex-link $n(1) $n([expr [expr $i + $servidores] + 3]) \
            [expr $BW * 100000] [expr $retardo * 0.001] DropTail
    } else {
        $ns duplex-link $n(2) $n([expr [expr $i + $servidores] + 3]) 1000Mb 1ms
DropTail
    }
}

#----- Listado de clientes
$ns set fue_ "";
for {set i 0} {$i < $clientes} {incr i} {
    $ns set fue_ "[ $ns set fue_] [list [expr [expr $i + $servidores] + 3]]"
}

#----- Configuración de la pagina web
set pool [new PagePool/WebTraf]

$pool set-num-client [llength [ $ns set fue_]]
set i 0
foreach s [ $ns set fue_] {
    $pool set-client $i $n($s)
    incr i
}

$pool set-num-server 1
$pool set-server 0 $n(4)
$pool set-num-session $sesiones

#----- SMTP
set i 0
foreach s [ $ns set fue_] {
#----- Conexión TCP
    set tcpi($i) [new Agent/TCP/Newreno]
    $ns attach-agent $n($s) $tcpi($i)
    set tcpsinki($i) [new Agent/TCPSink/DelAck]
    $ns attach-agent $n(3) $tcpsinki($i)
    $ns connect $tcpi($i) $tcpsinki($i)

    set tcpo($i) [new Agent/TCP/Newreno]
    $ns attach-agent $n(3) $tcpo($i)
    set tcpsinko($i) [new Agent/TCPSink/DelAck]
    $ns attach-agent $n($s) $tcpsinko($i)
    $ns connect $tcpo($i) $tcpsinko($i)
#----- Telnet
    set telnet($i) [new Application/Telnet]
}
}

```

```

$stelnet($i) attach-agent $tcpi($i)
$stelnet($i) set interval_ [expr [$intervalo value]]
$ns at 0.1 "$stelnet($i) start"
$ns at $tiempo "$stelnet($i) stop"
#----- FTP
set t 0
set j 0
while {$t < $tiempo} {
    set ftpi($i,$j) [$tcpi($i) attach-source FTP]
    set t [expr $t + [$tiempoftpi value]]
    set archivo [expr [$tamanoftpi value]]
    $ns at $t "$ftpi($i,$j) send $archivo"
    incr j
}
set t 0
set j 0
while {$t < $tiempo} {
    set ftpo($i,$j) [$tcpo($i) attach-source FTP]
    set t [expr $t + [$tiempoftpo value]]
    set archivo [expr [$tamanoftpo value]]
    $ns at $t "$ftpo($i,$j) send $archivo"
    incr j
}
#-----
incr i
}

#***** HTTP
#----- Páginas secundarias
set t 0
set i 0
while {$t < $tiempo} {
    set numpag [uniform 2 6] ;# Número consultas
    set interpag [new RandomVariable/Exponential]
    $interpag set avg_ 10 ;# Tiempo entre consultas
    set tampag [new RandomVariable/Constant]
    $stampag set val_ 8 ;# Cantidad de objetos
    set interobj [new RandomVariable/Exponential]
    $interobj set avg_ 0.01 ;# Tiempo entre envío de objetos
    set tamobj [new RandomVariable/ParetoII]
    $tamobj set avg_ 5 ;# Número de paquetes del objeto
    (1kB)
    $tamobj set shape_ 1.5
    $pool create-session $i $numpag [expr $t + 0.1] $interpag $stampag $interobj
    $tamobj
    set t [expr $t + [$intersesion0 value]]
    incr i
}
#----- Página principal
set t 0
while {$t < $tiempo} {
    set interpag [new RandomVariable/Constant]
    $interpag set val_ 0 ;# Tiempo entre consultas
    set tampag [new RandomVariable/Constant]
    $stampag set val_ 80 ;# Cantidad de objetos
    set interobj [new RandomVariable/Exponential]
    $interobj set avg_ 0.01 ;# Tiempo entre envío de objetos
    set tamobj [new RandomVariable/ParetoII]
    $tamobj set avg_ 5 ;# Número de paquetes del objeto
    (1kB)
    $tamobj set shape_ 1.5

```

```

    $pool create-session $i 1 [expr $t + 0.1] $interpag $stampag $interobj $tamobj
    set t [expr $t + [$intersesion1 value]]
    incr i
}

#***** Final simulación
$ns at $tiempo "finalizar"

proc finalizar {} {
    global ns
    $ns flush-trace
    exec nam uislan.nam &
    exit 0
}

puts "Escribiendo el archivo *.out. Esto puede tomar varios minutos..."
$ns run

```

C.2 Código para obtener el grano deseado del simulador

#Previo a utilizar este código, utilice el comando grep para filtrar la #información correspondiente al segmento que quiere estandarizar en tiempo.

```

# grep "^r" nombre_archivo | grep "nodotx nodorx tcp" > nuevo_archivo
# ejemplo: grep "^r" uislan.out | grep "0 2 tcp" > temp.out

```

#donde, el trafico va desde nodotx hasta nodorx

```

#Luego digite la siguiente línea para ejecutar este programa:
#perl programa.pl archivo_grep.out tiempo > archivo_final.txt
#ejemplo: perl jag.pl temp.out 0.625 > simulacion1.txt

```

```

$archivo = $ARGV[0];
$grano = $ARGV[1];

open (DATA,"<$archivo") || die ("no existe el archivo $!");

$s = 0;
#$stotal = 0;
$t = $grano;

while (<DATA>) {
    @x = split(' ');
    if ($x[1] < $t) {
        $s = $s + $x[5];
    }
    else {
        $s = $s/1000;
        print STDOUT "$s \n";
#        $stotal = $stotal + $s;
        $s = 0;
        $t = $t + $grano;
    }
}

$s = $s/1000;
# $stotal = $stotal + $s;

```

```
#      $stotal = $stotal/($t - 1);  
#      print STDOUT "$s ";  
print STDOUT "$s";  
  
close DATA;  
exit(0);
```